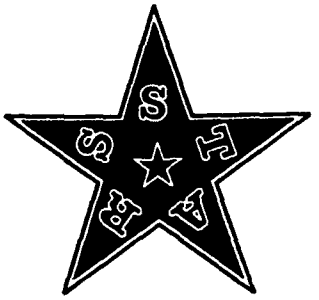
SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS



STARS Catalog

October 9, 1992

ASSET
2611 Cranberry Square
Morgantown, WV 26505
(304) 594-3954
librarian@source.asset.com

Asset Source for Software Engineering Technology

TABLE OF CONTENTS

Introduction....1
What Is STARS?....1
Ordering Instructions....2
Ada Standards And Bindings....3
Data Base Management System....4
Operating Systems....6
Reuse Library....6
Software Development Process....11
Software Documenting....21
Software Engineering Environment....22
Software Reuse....32
STARS Program Management....35
Text Processing....37
User Interface....38
Index....41

DISCLAIMER

Introduction

The STARS program has prepared this catalog to help interested parties identify useful STARS key products. The types of key products available include technical reports, software and related documentation.

The STARS Catalog is organized by categories as listed in the table of contents. Each category is delineated by a general title followed by related asset listings. Some assets may appear under more than one category. Each listing contains a brief description of the asset including its specific attributes. These attributes consist of: Order Number, a unique identifier for each asset; Version, indicating the current release of the asset available from the ASSET library; Release Date, the date this version of the asset became available; Producer, the name of the organization that developed the asset; Reference, a listing of additional identification numbers; Asset Type, a description of the asset as software, documentation, etc..; Domains, a listing of all the categories where the asset appears; Keyword, identifying the focus of the asset; Distribution, indicating restrictions that have been placed on the asset's availability.

What is STARS?

The Software Technology for Adaptable, Reliable Systems (STARS) Program is sponsored by DARPA, contracted through Air Force Electronic System Division and involves three cooperating Primes - Boeing, IBM, and Paramax - and a large number of subcontractors. The STARS mission statement is to increase software productivity, reliability and quality by synergistically integrating support for modern software development processes and modern reuse concepts within state of the art software engineering environment technology. STARS is focused on accelerating a change in the way software is developed within the DoD.

Ordering Instructions

You can obtain copies of the items in this catalog from the Asset Source for Software Engineering Technology (ASSET), a U. S. Department of Defense project to promote the reuse of computer software and software-related assets. ASSET's facility in Morgantown, West Virginia, supplies computer access to software reuse libraries, catalogs, and information via wide area networks and telecommunications. An order form and ASSET account application are located at the back of this document.

ASSET
2611 Cranberry Square
Morgantown, WV 26505
(304) 594-1762
FAX (304) 594-3951
Internet: librarian@source.asset.com

You can obtain hardcopies of the items in this catalog from the National Technical Information Service (NTIS) or from the Defense Technical Information Center (DTIC). When ordering, be sure to include the title of each item, together with its DTIC Accession Document (AD) number.

NTIS only has items which are approved for public release. Items can be ordered through the mail or over the phone. A variety of payment options are available.

NTIS
5285 Port Royal Road
Springfield, VA 22161-0001
(800) 553-NTIS (553-6847)
(703) 487-4650 in Washington, DC

To obtain items from DTIC, you must be a registered DTIC user. Please contact DTIC for details.

Defense Technical Information Center
Cameron Station
Alexandria, VA 22304-6145
(703) 274-7633

ADA STANDARDS AND BINDINGS

ADA PCTE BINDING (ADAPCTE)

This is the Paramax Ada binding to PCTE. The specification of the Ada binding is based on the ECMA (European Computer Manufacturers Association) Ada PCTE specification (Standard ECMA-162 Ada Language Binding, December 1991). The binding uses the GIE Emeraude PCTE 1.5 version 12.2 C libraries. Some changes to the ECMA specifications were made to accommodate the binding to PCTE 1.5. The release includes the AdaPCTE Version Description Document, which describes this version of AdaPCTE and provides installation instructions. These documents are provided in both plain ASCII and PostScript forms.

Order Number: ASSET_A_257
Alternate Name: ADAPCTE
Version: 0.1
Release Date: 12-JUN-92
Producer: PARAMAX
Reference: CDRL 04014, STARS-TC-04014/001/00
Asset Type: SOFTWARE CODE
Domains: ADA STANDARDS AND BINDINGS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: BINDINGS, PCTE, PORTABLE COMMON TOOL ENVIRONMENT
Distribution: Distribution authorized to US Government agencies and their contractors

ADA SEMANTIC INTERFACE SPECIFICATION (ASIS)

The Ada Semantic Interface Specification is a layered vendor independent open architecture. ASIS queries and services provide a consistent interface to information within the Ada Program Library. Clients of ASIS are shielded and free from the implementation details of each Ada vendor's proprietary library and intermediate representations.

This document consists solely of Ada package (design) specifications with no accompanying software or other documentation.

Order Number: ASSET_A_313
Version: 0.4
Release Date: 21-OCT-91
Producer: PARAMAX
Reference: CDRL 251901-003, DTIC
Asset Type: SOFTWARE DOCUMENTATION
Domains: ADA STANDARDS AND BINDINGS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: ADA COMPILER, ADA PROGRAM LIBRARY, ADA SEMANTIC INFORMATION, ASIS, INTERFACE, OPEN ARCHITECTURE, VENDOR-INDEPENDENT
Distribution: Approved for public release, distribution is unlimited

ADA/SQL BINDINGS

The ANSI Standard binding of SQL to Ada (ANSI X3.168) specifies how Ada programs may access the services of relational databases which conform to ANSI Standard SQL (ANSI X3.135). The ANSI binding uses weak, primitive data types, but most application programs need a binding which deals with strong, user-defined types. To use the ANSI binding in such circumstances, an "abstract module" must be placed between the application program and the ANSI binding in order to raise the level of abstraction of the binding. A specification for such an abstract module is provided by "Guidelines for Use of the SAME," Software Engineering Institute Technical Report CMU/SEI-89-TR-16. ("SAME" is an acronym for "SQL/Ada Module Extensions.") Unfortunately, applying the SAME method without any automated support would be a tedious task. "Ada/SQL Bindings" provides the needed automated support. It presumes that an implementation of the module version of the ANSI Ada/SQL binding is provided by the database vendor. It then permits a user to generate an abstract module conforming to the SAME method by instantiating Ada generics and executing the resulting program.

Documents supporting this software are available. They are "General Definition of Project" (ASSET_A_280; DTIC # AD-A228481), and "User's Manual for a Prototype Binding of ANSI-Standard SQL to Ada Supporting the SAME Methodology" (ASSET_A_281; DTIC # AD-A228480). They are available from ASSET in paper format only.

Order Number: ASSET_A_159
Alternate Name: PROTOTYPE BINDING OF ANSI-STANDARD SQL TO ADA SUPPORTING THE SAME METHODOLOGY
Version: 1
Release Date: 14-JUN-90
Producer: IBM CORPORATION
Reference: CDRL 02010-001A, DTIC AD-M000059
Asset Type: SOFTWARE CODE
Domains: ADA STANDARDS AND BINDINGS, DATA BASE MANAGEMENT SYSTEM
Keywords: DBMS, SAME
Distribution: Approved for public release, distribution is unlimited

ADA/SQL BINDINGS: GENERAL DEFINITION OF PROJECT

This report presents the background, technical approach, and top-level capabilities of the project to implement an Ada binding to SQL (Structured Query Language). It also discusses such technical problems as storing arbitrary data types in a data base and using SAME (SQL/Ada Module Extensions) without a module language compiler. Ada code is available (ASSET_A_159; Ada/SQL Bindings. DTIC number AD-M000059) to automatically generate the Ada package specifications and bodies to support the SAME Methodology. A User's Manual (ASSET_A_281: User's Manual for a Prototype Binding of ANSI-Standard SQL to Ada Supporting the SAME Methodology. DTIC # AD-A228480) is also available containing installation instructions, compilation order, guidelines for input data, detailed steps to create a specific binding, and information about porting the system to another DBMS.

This Ada/SQL document is available from ASSET in paper copy only.

Order Number: ASSET_A_280
Alternate Name: GENERAL DEFINITION OF PROJECT (ADA/SQL BINDING)
Version: 1
Release Date: 31-DEC-89
Producer: IBM CORPORATION
Reference: CDRL 02000-001, DTIC AD-A228481
Asset Type: SOFTWARE DOCUMENTATION
Domains: ADA STANDARDS AND BINDINGS, DATA BASE MANAGEMENT SYSTEM
Keywords: DBMS, SAME
Distribution: Approved for public release, distribution is unlimited

ADA/SQL BINDINGS: USERS MANUAL

This User's Manual for the Ada/SQL Prototype Binding contains installation instructions, compilation order, guidelines for input data, detailed steps to create a specific binding, and information about porting the system to another DBMS. The Ada software code is available (ASSET_A_159: Ada/SQL Bindings. DTIC # AD-M000059) to automatically generate the Ada package specifications and bodies to support the SAME methodology. Also available is a document, "General Definition of Project" (ASSET_A_280; DTIC # AD-A228481) which presents the background, technical approach, and top-level capabilities of the project.

This document is available from ASSET in paper copy only.

User's Manual for a Prototype Binding of ANSI-Standard SQL to Ada Supporting the SAME Methodology.

Order Number: ASSET_A_281
Alternate Name: USERS MANUAL FOR A PROTOTYPE BINDING OF ANSI-STANDARD SQL TO ADA
Version: 1
Release Date: 30-JUN-90
Producer: IBM CORPORATION
Reference: CDRL 02020-001, DTIC AD-A228480
Asset Type: SOFTWARE DOCUMENTATION
Domains: ADA STANDARDS AND BINDINGS, DATA BASE MANAGEMENT SYSTEM
Keywords: DBMS, SAME
Distribution: Approved for public release, distribution is unlimited

DATA BASE MANAGEMENT SYSTEM

ADA REPORT PRODUCTION SYSTEM (ARPS)

The Ada Report Production System (ARPS) provides the ability to specify and automatically generate reports from a relational database. Reports are created, maintained, and edited using an interactive report specification tool which provides for selection and specification of standard report formatting features and for database extraction using the Ada/SQL RDBMS interface. The report specification developed using this interactive tool is an Ada specification which defines, in Ada syntax, the format and contents of the desired report. The Ada report specification is processed by the ARPS code generator to produce Ada source code (packages) implementing the report formats and providing procedure calls and interfaces to the Ada/SQL for eventual report production. ARPS uses the X Window system and runs on a SUN 3/60 on UNIX.

Order Number: ASSET_A_158
Alternate Name: ADA SOURCE CODE - ADA REPORT PRODUCTION SYSTEM (ARPS)
Version: 2.0
Release Date: 17-MAR-89
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BQ13-023, CDRL 0530, DTIC AD-A240473
Asset Type: SOFTWARE CODE
Domains: DATA BASE MANAGEMENT SYSTEM
Keywords: DBMS
Distribution: Distribution authorized to US Government agencies and their contractors

ADA REPORT PRODUCTION SYSTEM (ARPS) - TECHNICAL REPORT

This document provides an overview of the Ada Report Production System (ARPS). It notes the enhancements that were made as a result of going from version 1.0 to 2.0. It also provides pointers to finding supporting documentation and the source code itself.

Order Number: ASSET_A_279
Alternate Name: TECHNICAL REPORT - ADA REPORT PRODUCTION SYSTEM (ARPS)

Version: 2.0
Release Date: 10-MAR-89
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BQ13-024, CDRL 0540, DTIC AD-B158022
Asset Type: SOFTWARE DOCUMENTATION
Domains: DATA BASE MANAGEMENT SYSTEM
Keywords: ARPS, DBMS
Distribution: Distribution authorized to US Government agencies and their contractors

ADA/SQL BINDINGS

The ANSI Standard binding of SQL to Ada (ANSI X3.168) specifies how Ada programs may access the services of relational databases which conform to ANSI Standard SQL (ANSI X3.135). The ANSI binding uses weak, primitive data types, but most application programs need a binding which deals with strong, user-defined types. To use the ANSI binding in such circumstances, an "abstract module" must be placed between the application program and the ANSI binding in order to raise the level of abstraction of the binding. A specification for such an abstract module is provided by "Guidelines for Use of the SAME," Software Engineering Institute Technical Report CMU/SEI-89-TR-16. ("SAME" is an acronym for "SQL/Ada Module Extensions.") Unfortunately, applying the SAME method without any automated support would be a tedious task. "Ada/SQL Bindings" provides the needed automated support. It presumes that an implementation of the module version of the ANSI Ada/SQL binding is provided by the database vendor. It then permits a user to generate an abstract module conforming to the SAME method by instantiating Ada generics and executing the resulting program.

Documents supporting this software are available. They are "General Definition of Project" (ASSET_A_280; DTIC # AD-A228481), and "User's Manual for a Prototype Binding of ANSI-Standard SQL to Ada Supporting the SAME Methodology" (ASSET_A_281; DTIC # AD-A228480). They are available from ASSET in paper format only.

Order Number: ASSET_A_159
Alternate Name: PROTOTYPE BINDING OF ANSI-STANDARD SQL TO ADA SUPPORTING THE SAME METHODOLOGY
Version: 1
Release Date: 14-JUN-90
Producer: IBM CORPORATION
Reference: CDRL 02010-001A, DTIC AD-M000059
Asset Type: SOFTWARE CODE
Domains: ADA STANDARDS AND BINDINGS, DATA BASE MANAGEMENT SYSTEM
Keywords: DBMS, SAME
Distribution: Approved for public release, distribution is unlimited

ADA/SQL BINDINGS: GENERAL DEFINITION OF PROJECT

This report presents the background, technical approach, and top-level capabilities of the project to implement an Ada binding to SQL (Structured Query Language). It also discusses such technical problems as storing arbitrary data types in a data base and using SAME (SQL/Ada Module Extensions) without a module language compiler. Ada code is available (ASSET_A_159; Ada/SQL Bindings. DTIC number AD-M000059) to automatically generate the Ada package specifications and bodies to support the SAME Methodology. A User's Manual (ASSET_A_281: User's Manual for a Prototype Binding of ANSI-Standard SQL to Ada Supporting the SAME Methodology. DTIC # AD-A228480) is also available containing installation instructions, compilation order, guidelines for input data, detailed steps to create a specific binding, and information about porting the system to another DBMS.

This Ada/SQL document is available from ASSET in paper copy only.

Order Number: ASSET_A_280
Alternate Name: GENERAL DEFINITION OF PROJECT (ADA/SQL BINDING)
Version: 1
Release Date: 31-DEC-89
Producer: IBM CORPORATION
Reference: CDRL 02000-001, DTIC AD-A228481
Asset Type: SOFTWARE DOCUMENTATION
Domains: ADA STANDARDS AND BINDINGS, DATA BASE MANAGEMENT SYSTEM
Keywords: DBMS, SAME
Distribution: Approved for public release, distribution is unlimited

ADA/SQL BINDINGS: USERS MANUAL

This User's Manual for the Ada/SQL Prototype Binding contains installation instructions, compilation order, guidelines for input data, detailed steps to create a specific binding, and information about porting the system to another DBMS. The Ada software code is available (ASSET_A_159: Ada/SQL Bindings. DTIC # AD-M000059) to automatically generate the Ada package specifications and bodies to support the SAME methodology. Also available is a document, "General Definition of Project" (ASSET_A_280; DTIC # AD-A228481) which presents the background, technical approach, and top-level capabilities of the project.

This document is available from ASSET in paper copy only.

User's Manual for a Prototype Binding of ANSI-Standard SQL to Ada Supporting the SAME Methodology.

Order Number: ASSET_A_281
Alternate Name: USERS MANUAL FOR A PROTOTYPE BINDING OF ANSI-STANDARD SQL TO ADA
Version: 1
Release Date: 30-JUN-90
Producer: IBM CORPORATION

Reference: CDRL 02020-001, DTIC AD-A228480
Asset Type: SOFTWARE DOCUMENTATION
Domains: ADA STANDARDS AND BINDINGS, DATA BASE MANAGEMENT SYSTEM
Keywords: DBMS, SAME
Distribution: Approved for public release, distribution is unlimited

OPERATING SYSTEMS

ADA INTERFACES TO POSIX

This report considers the applicability of the Portable Operating System Interface (POSIX) to the development of the Software Engineering Environment (SEE) for the Software Technology for Adaptable, Reliable Systems (STARS) program. It compares and contrasts characteristics and potential overlap of object management interfaces (in this example, CAIS-A) and POSIX. Because there are overlaps in the objectives for POSIX and an object manager, there are some apparent overlaps in the functions provided by the interfaces. It is concluded that there is no actual overlap in function in the I/O model for persistent data nor in the execution control models. Interfaces like CAIS-A should be the portable interface set for tools, while POSIX offers a method for gaining portability for the CAIS and other interfaces in the base level of the SEE.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_155
Alternate Name: INTERFACE STANDARDS INFORMAL TECHNICAL DATA, ADA INTERFACES TO POSIX
Version: 1
Release Date: 14-APR-89
Producer: PARAMAX
Reference: CDRL 02021-005, DTIC AD-A228820
Asset Type: DOCUMENT
Domains: OPERATING SYSTEMS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: COMPARISON REPORT, INTERFACE STANDARDS, POSIX
Distribution: Approved for public release, distribution is unlimited

MACH VS. UNIX PERFORMANCE REPORT

This report describes the study conducted to measure the performance of multiprogramming applications running under MACH with similar applications running under UNIX -- using CAIS-A tools as the test vehicle. Specifically, the study compared the elapsed times for a large set of CAIS-A interface calls made between CAIS-A tools and the CAIS-A server program. The report discusses the differences in implementation approaches used for message passing between tools the server. It describes the approach taken to gather the performance data. The bulk of the report analyzes the data gathered, which showed that most interfaces were 10 to 30 percent faster under MACH.

Order Number: ASSET_A_172
Version: 1
Release Date: 30-JUL-90
Producer: PARAMAX
Reference: CDRL 00900, DTIC AD-B152497
Asset Type: DOCUMENT
Domains: OPERATING SYSTEMS
Keywords: MACH, PERFORMANCE, UNIX
Distribution: Approved for public release, distribution is unlimited

REUSE LIBRARY

AMS (ASSET MANAGEMENT SYSTEM) PRODUCT DEFINITION DOCUMENT

The Product Definition Document for the Asset Management System specifies the functional capabilities, inherent concepts, external interfaces, operational concept for use, and development methodology for the Asset Management System reuse library mechanism.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_289
Alternate Name: PRODUCT DEFINITION DOCUMENT FOR ASSET MANAGEMENT SYSTEM (AMS)
Version: 1
Release Date: 27-APR-91
Producer: IBM CORPORATION
Reference: CDRL 03036-001, DTIC
Asset Type: DOCUMENT
Domains: REUSE LIBRARY
Keywords: AMS, ARCS, ASSET MANAGEMENT SYSTEM, INQUISIX, LIBRARY, REUSE LIBRARY MECHANISM
Distribution: Approved for public release, distribution is unlimited

ASSET CATALOG

This Catalog is a snapshot of the contents of the ASSET Reuse Library as of the release date. The reusable software components contained in the Library are listed by domain. Besides a description of all the currently cataloged assets, the user will find an ASSET account application form and an asset order form.

Order Number: ASSET_A_260
Alternate Name: ASSET SOURCE FOR SOFTWARE ENGINEERING TECHNOLOGY (ASSET) CATALOG
Version: 92.2
Release Date: 09-OCT-92

Producer: IBM CORPORATION
Reference: CDRL 05604-001
Asset Type: CATALOG
Domains: REUSE LIBRARY
Keywords: CATALOG, REPOSITORY, SOFTWARE REUSE
Distribution: Approved for public release, distribution is unlimited

ASSET LIBRARY OPEN ARCHITECTURE FRAMEWORK (ALOAF)

The STARS Asset Library Open Architecture Framework (ALOAF) facilitates the exchange of reusable assets among diverse libraries and facilitates the construction of reuse tools that are portable between asset libraries. The asset interchange objective focuses the STARS ALOAF upon the information needed to systematically organize and describe assets stored within an asset library. The ALOAF addresses the interchange of assets and their associated asset descriptions and model information through the ALOAF Data Modeling and Asset Interchange Specification. The portable reuse tools objective focuses the STARS ALOAF upon the asset library services and standard interfaces needed by reuse-based library tools. The ability to create portable reuse-based library tools is addressed by the ALOAF Service Model along with their ALOAF Programmatic Interfaces.

Asset interchange and asset service interfaces are critical elements in achieving a broader objective - asset libraries which interoperate to such an extent that the boundaries between individual libraries become invisible to the end user. In general terms, this is the STARS vision of "seamless" library interoperation.

Order Number: ASSET_A_251
Alternate Name: ALOAF
Version: 1.2
Release Date: 14-AUG-92
Producer: PARAMAX
Reference: CDRL 04041
Asset Type: DOCUMENT
Domains: REUSE LIBRARY
Keywords: INTERFACE STANDARDS, INTEROPERABILITY, LIBRARY MANAGEMENT
Distribution: Approved for public release, distribution is unlimited

CDD/REPOSITORY ENACTMENT MECHANISM DEVELOPMENT

This document is one of a group describing features of process enactment of the Boeing CDD/Repository based environment. This document describes the specification, design, and type extensions made for the process mechanism to enable process enactment in the CDD/Repository based environment.

This document also describes the software that implements the mechanism, the installation and testing of the software, and usage of the mechanism. A demonstration example using the process enactment mechanism to enable process enactment is described in Boeing CDRL 4027R-2, (ASSET_A_303: Process Enactment Demonstration).

This document is available from ASSET in PostScript format only, or on paper.

Order Number: ASSET_A_302
Alternate Name: STARS ENACTMENT MECHANISM DEVELOPMENT ON CDD/REPOSITORY
Version: 1
Release Date: 24-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 4027R-1
Asset Type: DOCUMENT
Domains: REUSE LIBRARY, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: COHESION, CONTROL INTEGRATION, DATA INTEGRATION, PROCESS ENACTMENT, PROCESS MODELING, SOFTWARE ENGINEERING ENVIRONMENT, STARS
Distribution: Distribution authorized to US Government agencies and their contractors

COMMAND CENTER DOMAIN MODEL DESCRIPTION - CARDS

This document describes Version 1.0 of the library model for the Central Archive for Reusable Defense Software (CARDS) Command Center (CC) Library. A CARDS library model for a domain-specific reuse library is a formal encoding of information produced during domain engineering activities. The purpose of a CARDS domain-specific library model is to: capture critical information such as domain requirements and generic architectures that is produced by domain engineering activities; based on this information, describe criteria for qualification and insertion of reusable assets into the library; provide a basis for organizing ("classifying") reusable assets for search and retrieval applications; and provide a basis for constructing other kinds of reuse library applications. Version 1.0 of the CC Library model encodes a "snapshot" of the Generic Command Center Architecture (GCCA) and related products produced by the Portable Reusable Integrated Software Modules (PRISM) program, as these products existed prior to April 1992. The formalism used for this encoding is provided by the Reusability Library Framework (RLF).

Order Number: ASSET_A_270
Version: 1
Release Date: 02-JUN-92
Producer: PARAMAX
Reference: CDRL 03534, STARS-AC-03534/001/00
Asset Type: DOCUMENT
Domains: REUSE LIBRARY
Keywords: DOMAIN-SPECIFIC, RLF
Distribution: Distribution authorized to US Government agencies and their contractors

COMMAND CENTER LIBRARY - TECHNICAL CONCEPT

This document describes the technical concepts of the command center (CC) library, including library modeling, the library software infrastructure, security, and interoperability. This document will baseline the technical foundation for the CC library and for other domain-specific libraries to be implemented by the Central Archive for Reusable Defense Software (CARDS) program.

Order Number: ASSET_A_269
Alternate Name: TECHNICAL CONCEPT - COMMAND CENTER LIBRARY
Version: 1
Release Date: 25-AUG-92
Producer: PARAMAX
Reference: CDRL 04107A, STARS-AC-04107A/001/00
Asset Type: DOCUMENT
Domains: REUSE LIBRARY
Keywords: CARDS, COMMAND CENTER LIBRARY, DOMAIN-SPECIFIC, INTEROPERABILITY
Distribution: Distribution authorized to US Government agencies and their contractors

PRACTICAL ASPECTS OF REPOSITORY OPERATIONS

This document describes the operation of the STARS Repository computer and the basis for the policies under which it operated. The STARS Repository acted as a focal point of the STARS program and supported many STARS activities. It stored all deliverable software and technical reports. It was also a means of communication amongst the prime contractors, the subcontractors, the consultants, and the contracting agency. Electronic mail was interchanged, other electronic documents were shared, software was developed, and software was shared and reused by the STARS contract participants using the STARS Repository computer.

Order Number: ASSET_A_207
Version: 1
Release Date: 10-JAN-90
Producer: IBM CORPORATION
Reference: CDRL 01440-001, DTIC AD-A228455
Asset Type: DOCUMENT
Domains: REUSE LIBRARY, STARS PROGRAM MANAGEMENT
Keywords: GUIDEBOOK, OPERATIONS
Distribution: Approved for public release, distribution is unlimited

REPOSITORY GUIDELINES AND STANDARDS

This document recommends specific ways to collect, store, manage, select, and retrieve reusable software. It defines guidelines and standards for the contents of a software reuse library, and proposed tools to enforce these standards. It outlines the criteria that reuse library processes and guidelines must meet. Also addressed are the underlying reuse library concepts. STARS Reusability Guidelines, (ASSET_A_209, DTIC AD-A228468), contains a complementary set of guidelines that tell how to develop or alter components so they can be reused.

Order Number: ASSET_A_210
Version: 1
Release Date: 7-MAR-89
Producer: IBM CORPORATION
Reference: CDRL 00460-001, DTIC AD-A228484
Asset Type: DOCUMENT
Domains: REUSE LIBRARY
Keywords: LIBRARY MANAGEMENT, REPOSITORY
Distribution: Approved for public release, distribution is unlimited

REPOSITORY OPERATIONS AND PROCEDURES

An assessment of reuse library data focusing on issues of ownership, data rights, ITAR restrictions, as well as quality and consistency. The document also proposes terms and conditions for Government license agreements and provides sample forms to collect data for operating a reuse library. The focus is on the policies and procedures for the STARS Repository.

Order Number: ASSET_A_211
Version: 1
Release Date: 7-MAR-90
Producer: IBM CORPORATION
Reference: CDRL 01470-001, DTIC AD-A228472
Asset Type: DOCUMENT
Domains: REUSE LIBRARY
Keywords: DATA RIGHTS, OPERATIONS, OWNERSHIP, RESTRICTION
Distribution: Approved for public release, distribution is unlimited

REPOSITORY SPECIFICATIONS

A specification of the capabilities, characteristics and tools for a software reuse library. Defines the minimum set of integrated software development and data processing facilities needed to maintain and support a software reuse library.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_175
Alternate Name: REPOSITORY PROTOTYPE SYSTEM SPECIFICATION
Version: 1
Release Date: 16-FEB-90
Producer: IBM CORPORATION
Reference: CDRL 01590-001, DTIC AD-A228467
Asset Type: DOCUMENT
Domains: REUSE LIBRARY
Keywords: REPOSITORY SPECIFICATION
Distribution: Approved for public release, distribution is unlimited

REUSABILITY LIBRARY FRAMEWORK

The Reusability Library Framework (RLF) is a software system designed and implemented to enable the development of application-area (or domain) specific reuse libraries. The framework has been implemented in Ada, using basic principles of software engineering, and is both extensible and integratable with other Ada systems. The RLF includes a semantic network subsystem and a rule-driven inferencing subsystem which are used to record domain knowledge and heuristics in a form suitable for use within Ada applications. Besides the creation of several kinds of librarian applications, the RLF has been used to represent process model descriptions, capture Ada and heuristic testing information for use by an Ada unit test plan generator, and to actively aid a documentation specialist in producing structured technical documentation from templates and a knowledge-based model of the document production software being used.

Order Number: ASSET_A_264
Alternate Name: RLF
Version: 3.0
Release Date: 23-JAN-92
Producer: PARAMAX
Reference: CDRL 03064, STARS-TC-03064/004/00
Asset Type: SOFTWARE CODE
Domains: REUSE LIBRARY
Keywords: DOMAIN SPECIFIC
Distribution: Approved for public release, distribution is unlimited

REUSE LIBRARY PROCESS MODEL

This delivery of the STARS Reuse Library Process Model contains the Domain Analysis Process building block. Domain analysis is defined as the process by which information used in developing software systems is identified, captured, and organized with the purpose of making it reusable. The objective of the process model is to convert the ad hoc nature of domain analysis into a repeatable procedure with well-defined, tangible outputs.

Order Number: ASSET_A_177
Version: 1
Release Date: 26-JUL-91
Producer: IBM CORPORATION
Reference: CDRL 03041-002, DTIC AD-B157091
Asset Type: DOCUMENT
Domains: REUSE LIBRARY, SOFTWARE DEVELOPMENT PROCESS
Keywords: DOMAIN ANALYSIS
Distribution: Approved for public release, distribution is unlimited

REUSE LIBRARY PROTOTYPE

This Ada software is a prototype library mechanism which provides functions to search, browse, and extract reusable assets. It also allows for users to subscribe to assets, to supply new assets and to submit problems and evaluations of assets. It relies on an Oracle database engine for data storage, and was developed on the IBM RISC 6000 computer with the VERDIX compiler.

Order Number: ASSET_A_290
Alternate Name: STARS REUSE LIBRARY (SRL) PROTOTYPE
Version: 2.6
Release Date: 31-MAY-91
Producer: IBM CORPORATION
Reference: CDRL 03038-001
Asset Type: SOFTWARE CODE
Domains: REUSE LIBRARY
Keywords: LIBRARY MECHANISM, REUSE LIBRARY
Distribution: Approved for public release, distribution is unlimited

REUSE LIBRARY PROTOTYPE - LIBRARIANS GUIDE

This is the user's guide for the AIX RISC System/6000 version of the STARS Reuse Library Prototype, Version 2.6. It explains how to supply assets, search for assets, and extract assets.

[Note: This User's Guide does not apply to the ASSET Reuse Library.]

Order Number: ASSET_A_178
Alternate Name: USERS GUIDE FOR REUSE LIBRARY PROTOTYPE VERSION 2.6
Version: 2.6
Release Date: 27-MAY-91
Producer: IBM CORPORATION
Reference: CDRL 03037-001, DTIC AD-B157092
Asset Type: SOFTWARE DOCUMENTATION
Domains: REUSE LIBRARY
Keywords: GUIDEBOOK, REUSE LIBRARY
Distribution: Approved for public release, distribution is unlimited

RLF ADAKNET USERS MANUAL

AdaKNET (Ada Knowledge NETwork) is the semantic network subsystem of the RLF. AdaKNET enables the creation and modification of structured inheritance networks to represent detailed patterns of information. The manual outlines the current structure of the AdaKNET system and indicates how to make effective use of the available programmatic interfaces. A sample session description is included along with an appendix describing the basic information structuring primitives within AdaKNET. The manual also describes the specification language used to declare semantic networks for processing by AdaKNET.

Order Number: ASSET_A_266
Alternate Name: REUSABILITY LIBRARY FRAMEWORK ADAKNET USERS MANUAL
Version: 3.0
Release Date: 05-DEC-91
Producer: PARAMAX
Reference: CDRL 04046, STARS-TC-04046/002/00

Asset Type: SOFTWARE DOCUMENTATION
Domains: REUSE LIBRARY
Keywords: DOMAIN SPECIFIC, USERS GUIDE
Distribution: Approved for public release, distribution is unlimited

RLF ADATAU USERS MANUAL

AdaTAU is the inferencing subsystem included in the RLF. TAU stands for "Think, Ask, Update" which is a summary of the inferencing approach taken by AdaTAU. The manual describes how to install AdaTAU, provides a brief background for the current structure of the AdaTAU system and provides a list of the programmatic interfaces available within AdaTAU. Advice is given to users for configuring applications to make use of AdaTAU. A sample session showing the use of the included AdaTAU test harness is also included. The manual closes with a description of the specification language used to define inference bases.

Order Number: ASSET_A_267
Alternate Name: REUSABILITY LIBRARY FRAMEWORK ADATAU USERS MANUAL
Version: 3.0
Release Date: 05-DEC-91
Producer: PARAMAX
Reference: CDRL 04046, STARS-TC-04046/003/00
Asset Type: SOFTWARE DOCUMENTATION
Domains: REUSE LIBRARY
Keywords: DOMAIN SPECIFIC, USERS GUIDE
Distribution: Approved for public release, distribution is unlimited

RLF GRAPHICAL BROWSER USERS MANUAL

The RLF GB 3.0 User's Manual describes the use and basic customization of the Reusability Library Framework (RLF) Graphical Browser (GB), referred to as the RLF GB. The reader is not expected to be a programmer, but familiarity with the UNIX C Shell, UNIX files and directories, and basic X Window System operations using the TAB Window Manager (twm) is assumed. Some explanations of RLF concepts is provided, but only at an elementary level.

After the introduction, the manual provides tool invocation procedures, followed by sections that describe the operation of the RLF GB, which compose the bulk of the manual. There are appendices on customization, error messages and problem reporting procedures.

The RLF GB 3.0 User's Manual is for the RLF GB released as part of the RLF, Version 3.0, released in January, 1992. This release runs on SunOS, version 4.1.1 or later. The manual assumes a UNIX C shell interpreter is accessible to the user.

Order Number: ASSET_A_268
Alternate Name: REUSABILITY LIBRARY FRAMEWORK GRAPHICAL BROWSER USERS MANUAL
Version: 3.0
Release Date: 31-JAN-92
Producer: PARAMAX
Reference: CDRL 04046, STARS-TC-04046/004/00
Asset Type: SOFTWARE DOCUMENTATION
Domains: REUSE LIBRARY
Keywords: DOMAIN SPECIFIC, USERS GUIDE
Distribution: Approved for public release, distribution is unlimited

RLF LIBRARIAN USERS MANUAL

The Reusability Library Framework (RLF) is a software system designed and implemented to enable the development of application-area (or domain) specific reuse libraries. The RLF includes a semantic network subsystem and a rule-driven inferencing subsystem.

While librarians are not the only sorts of applications that have been built with the RLF, they are the applications that motivated the initial development of the RLF. This manual describes a demonstration librarian built for the domain of Ada benchmark programs. This application is typical of many RLF applications in that it uses a hybrid knowledge representation system incorporating an integrated form of AdaTAU and AdaKNET. The manual provides a librarian system overview and provides an annotated sample usage session. The manual also presents the hybrid knowledge base description language used to connect AdaKNET and AdaTAU.

Order Number: ASSET_A_265
Alternate Name: REUSABILITY LIBRARY FRAMEWORK LIBRARIAN USERS MANUAL
Version: 3.0
Release Date: 15-JAN-92
Producer: PARAMAX
Reference: CDRL 04046, STARS-TC-04046/001/00
Asset Type: SOFTWARE DOCUMENTATION
Domains: REUSE LIBRARY
Keywords: DOMAIN SPECIFIC, USERS GUIDE
Distribution: Approved for public release, distribution is unlimited

ROAMS TESTBED REPORT AND LESSONS LEARNED

The Boeing Defense & Space Group (DS&G) STARS Program designed and developed the Reusable Object Access and Management System (ROAMS), and integrated it with the system engineering environment (SEE) being developed by D&SG for the STARS program. ROAMS is a reuse library mechanism for an object-oriented, repository based software reuse library. This document describes the results and lessons learned during the ROAMS development and integration, and during ROAMS demonstrations with the SEE. Included are discussions of the ROAMS design history, major issues encountered and their resolutions, design issues on logical and physical reuse libraries, SEE environment features with navigators, element (object) editors, and the ROAMS extensions to the type hierarchy.

Order Number: ASSET_A_301
Alternate Name: REPORT ON RESULTS AND LESSONS LEARNED FROM ROAMS TESTBED OPERATIONS
Version: 1
Release Date: 22-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 1870
Asset Type: DOCUMENT
Domains: REUSE LIBRARY, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: ASSET, ATIS, LIBRARIAN, OBJECT-ORIENTED, RETRIEVE, REUSE LIBRARY, ROAMS, SLCSE, STARS, TAXONOMY
Distribution: Distribution authorized to US Government agencies and their contractors

SOFTWARE DEVELOPMENT PROCESS

ACE/EIS INTEGRATION LESSONS LEARNED

This document reports the lessons learned from integrating the Ada Command Environment (ACE) into the Prototype Engineering Information System (PREIS) environment. ACE is an interactive command language environment for Ada software development where Ada is both the programming language and the command language. PREIS is an object-oriented environment targeted for computer-aided engineering (CAE). During this task, PREIS demonstrated an ability to model several ACE constructs in an object-oriented fashion. ACE on the other hand, demonstrated its extensibility by the addition of abstract data types. These abstract data types give the user a window into the PREIS environment.

The ACE/PREIS integration effort was undertaken to identify and provide insights into issues related to tool integration in an object management system. The integration effort revealed the difficulty of integrating C with Ada and demonstrated again the necessity of good documentation. The requirement for a programmatic interface to PREIS for future integration efforts was also identified.

Order Number: ASSET_A_152
Version: 1
Release Date: 02-NOV-90
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BS20-002, CDRL 03041R, DTIC AD-B158007
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: ACE, EIS, FRAMEWORK, LESSONS LEARNED, SOFTWARE ENGINEERING ENVIRONMENT, TOOL INTEGRATION
Distribution: Approved for public release, distribution is unlimited

ADA COMMAND ENVIRONMENT

The Ada Command Environment (ACE) is an interactive Ada environment coupled with a set of Ada abstract data types (ADTs). The interactive environment allows users to rapidly prototype general Ada applications, while the ADTs allow prototyping of applications for particular domains, such as X Window System applications. In addition, the ADTs provide an Ada view of underlying applications, which when combined with the interactive environment replaces the traditional role of a command language. When using ACE, Ada becomes the command language as well as the programming language. This version of ACE includes support for X Window System prototyping. This asset includes the ACE Users Manual.

Order Number: ASSET_A_153
Alternate Name: ACE
Version: 8.0
Release Date: 29-Oct-1990
Producer: PARAMAX
Reference: CDRL 00990, DTIC AD-A228822, GR-7670-1141 (NP)
Asset Type: SOFTWARE CODE
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: COMMAND ENVIRONMENT, OBJECT-ORIENTED, PROGRAMMING-IN-THE-LARGE
Distribution: Approved for public release, distribution is unlimited

ADA COMMAND ENVIRONMENT USERS MANUAL

This technical report provides users of the Ada Command Environment (ACE) with a description of each of the packages and subprograms that are provided with ACE. In addition, the report describes the overall purpose of ACE, general guidelines on ACE usage, and provides examples of typical user interaction during an ACE session.

Order Number: ASSET_A_278
Alternate Name: ACE USERS MANUAL
Version: 8.0
Release Date: 29-OCT-90
Producer: PARAMAX
Reference: CDRL 00980, DTIC AD-A229400, STARS-RC-00980/001/00
Asset Type: SOFTWARE DOCUMENTATION
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: COMMAND ENVIRONMENT, OBJECT-ORIENTED, PROGRAMMING-IN-THE-LARGE

Distribution: Approved for public release, distribution is unlimited

ADA FORMAL METHODS IN THE STARS ENVIRONMENT

This report is a collection of several possible ways in which tools supporting formal methods might be made interoperable and/or integrated into the STARS SEE. The possibilities discussed are merely representative, and are limited only by the effort available for completing the report. In general, formal methods might be applied to any phase of system development, including requirements analysis, design, implementation of both hardware and software, testing, and so forth. However, this report is restricted for the most part to formal code verification of Ada. (The single exception to this restriction is the discussion of run-time checking in Anna.) Formal code verification is proof that an algorithm implemented by a piece of code meets a formally stated specification. ORA's Penelope and Ariel tools support code verification.

Order Number: ASSET_A_256
Version: 1
Release Date: 03-JUN-92
Producer: PARAMAX
Reference: CDRL 03089, STARS-SC-03089/001/00
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: FORMAL METHODS, SOFTWARE DEVELOPMENT TOOLS, STARS SEE, TOOL INTEGRATION
Distribution: Distribution authorized to US Government agencies and their contractors

ADA PROGRAM FLOW ANALYSIS TOOL (APFAT)

APFAT is an object-based, menu-driven program which produces the following reports: main indented call tree summary, subprogram invocations, subprogram references, exceptions usage summary, exceptions raised-by summary, identifier usage summary, user specified identifier usage summary, Booch diagrams, and program metrics. These reports may be directed to the user's screen or redirected to any output file.

The program is designed to process syntactically correct Ada source code in the correct compilation order. It runs on an Apollo system with UNIX under BSD4.2 and the Verdix Compiler SRS 3.5.1.

Order Number: ASSET_A_199
Alternate Name: APFAT
Version: 1
Release Date: 17-MAR-89
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BQ10-032, CDRL 0260-2, DTIC AD-B157893
Asset Type: SOFTWARE CODE
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: BOOCH DIAGRAM, CALLING TREE, METRICS
Distribution: Approved for public release, distribution is unlimited

ADA STATIC STRUCTURE EVALUATION TOOL (ASSET)

The Ada Static Structure Evaluation Tool (ASSET) is a port and enhancement of the SIMTEL 20 tool, "Compilation Order", (COMPORD). ASSET reads in Ada source code from a user-supplied list of files and writes dependency and compilation order reports either to standard output or to a user-specified file. ASSET can serve as a front end to another STARS tool, the Ada Program Flow Analysis Tool (APFAT).

Order Number: ASSET_A_202
Alternate Name: ASSET
Version: 2.0
Release Date: 17-MAR-89
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BQ10-34, CDRL 0260-3, DTIC AD-B157894
Asset Type: SOFTWARE CODE
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: ADA, COMPILATION ORDER
Distribution: Approved for public release, distribution is unlimited

AIR TRAFFIC CONTROL INFORMATION OBJECT MODEL

Information Object Modeling is a technique for developing specification models for systems. The techniques for building Information Object Models were adapted from techniques of real-time structured analysis and the Foxboro company's experience in specifying and developing real-time process control systems.

An Information Object Model (IOM) is organized to provide levels of information for different audiences, so that one document can meet the needs of different people. A mission statement is provided which describes the scope of the system. An overview of the system describes the major functional objects. Finally, each functional object is discussed in detail.

The modeling techniques for an IOM use the graphical techniques real-time structured analysis, including transformation diagrams (data flow plus control flow), state transition diagrams, and entity relationship diagrams. Transformation diagrams, however, are applied in a different manner, representing the communication of objects organized hierarchically rather than a functional decomposition of processes.

This document is a specification model for an air traffic control system prepared using Real Time Structured Analysis. It shows Foxboro's concept of specification packaging and can serve as an alternative to MIL-STD-2167A.

This document is available from ASSET only in SGML format, or on paper.

Order Number: ASSET_A_191
Alternate Name: STARS STRUCTURE (DOD AAS IOM DOCUMENT VERSION 1.3)
Version: 1.3
Release Date: 11-MAY-90
Producer: IBM CORPORATION
Reference: CDRL 01200-001, DTIC AD-A228479
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE DOCUMENTING
Keywords: REAL-TIME, STRUCTURED ANALYSIS
Distribution: Approved for public release, distribution is unlimited

APPLICATION BLUEPRINTS

Application blueprints serve as a framework for designing new systems in an application domain, leading to reuse of design information and greater reuse of code. This document defines the term application blueprint, tells how to create one, and discusses the benefits and drawbacks of this approach. The appendix presents a generic specification and information about the initial domain analysis for creating an application blueprint for an air traffic control system. This paper can be the basis for future research on reusing analysis and design information.

Order Number: ASSET_A_186
Alternate Name: UPDATED APPLICATION BLUEPRINT DEFINITION FOR C3
Version: 1
Release Date: 10-JAN-90
Producer: IBM CORPORATION
Reference: CDRL 01490-001A, DTIC AD-A228471
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE REUSE
Keywords: APPLICATION BLUEPRINTS, DOMAIN ANALYSIS
Distribution: Approved for public release, distribution is unlimited

CEPA DEVELOPMENT ACTIVITY LOG

This document provides the activity log of the development of the Cleanroom Engineering Process Assistant (CEPA) in the Knowledge-Integration Shell (KI Shell). This document records the activities performed in planning, designing and implementing the CEPA prototype system, and includes all relevant artifacts produced during the CEPA planning, design and development.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_285
Alternate Name: CLEANROOM ENGINEERING PROCESS ASSISTANT DEVELOPMENT IMPLEMENTATION LOG
Version: 1
Release Date: 30-SEP-91
Producer: IBM CORPORATION
Reference: CDRL 03705-001A, DTIC AD-B157269
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: CLEANROOM PROCESS
Distribution: Approved for public release, distribution is unlimited

CLEANROOM ENGINEERING DOCUMENT ABSTRACTS: STARS TASK IS-15

This document provides abstracts for the documentation artifacts produced as a result of STARS Task IS-15, Software Process Management.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_287
Alternate Name: STARS TASK IS-15: DOCUMENT ABSTRACTS
Version: 1
Release Date: 30-SEP-91
Producer: IBM CORPORATION
Reference: CDRL 03705-001C, DTIC AD-B157268
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: CLEANROOM PROCESS, STARS
Distribution: Approved for public release, distribution is unlimited

CLEANROOM ENGINEERING PROCESS ASSISTANT (CEPA) SPECIFICATIONS

This document provides the requirements for a process system to support the development of software using the "Cleanroom Engineering Software Development Process". This document provides functional specifications (in box structure notation) for the "Cleanroom Engineering Process Assistant (CEPA)" prototype system and provides a concept of operations for the CEPA support of the "Cleanroom Engineering Software Development Process".

This document is a companion document to the "Cleanroom Software Process Case Study" problem document (ASSET_A_273; IBM CDRL 03706-001) and the "Cleanroom Software Process Case Study: Executive Summary" (ASSET_A_188; IBM CDRL 03706-001B).

Order Number: ASSET_A_274
Alternate Name: CEPA SPECIFICATIONS FOR A PROTOTYPE WORKSTATION AND DISPATCHER
Version: FINAL
Release Date: 26-JUL-91

Producer: IBM CORPORATION
Reference: CDRL 03706-001A, DTIC AD-B157202
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: CASE STUDY, CLEANROOM ENGINEERING, KI SHELL, SOFTWARE PROCESS, SOFTWARE PROCESS ENACTMENT SUPPORT TOOLS
Distribution: Approved for public release, distribution is unlimited

CLEANROOM ENGINEERING REQUIREMENTS FOR PLANNING SUPPORT

This report presents the requirements for planning and scheduling for Cleanroom Engineering. The lessons learned/advances made as a result of the research effort include:

- The generic Cleanroom process model was improved.
- The full recognition of the fact that process definition must drive a process improvement program.
- A classification of all decisions necessary to support a project. A definition of planning requirements to support each type of decision.
- The development of an analytic method to compute expected project completion time using Markov model of project status.
- The recognition that the role of process managers (like CEPA, KI-Shell or Process Weaver) is to support the dispatching decision.
- The most effective way to support the dispatching decision is to centralize the necessary process intelligence in a process server.
- Full definition of the CEPA requirements to support project management and planning.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_291
Alternate Name: REQUIREMENTS FOR WBS TO SUPPORT MGMT OF SW DEV PROJ USING CLEANROOM ENGINEERING
Version: FINAL
Release Date: 31-JUL-92
Producer: IBM CORPORATION
Reference: CDRL 04024-002A, DTIC AD-B167448
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: CLEANROOM ENGINEERING, PLANNING, PROJECT MANAGEMENT, SCHEDULING, SOFTWARE DEVELOPMENT, SOFTWARE PROCESS
Distribution: Distribution authorized to US Government agencies and their contractors

CLEANROOM ENGINEERING SOFTWARE DEVELOPMENT PROCESS

Cleanroom provides software developers with the basis for developing software under statistical quality control. Software is functionally verified and certified, not tested, using sampling techniques, thus permitting software developers to assert a mean time to failure (MTTF) for the software modules they develop.

Under the STARS Program, the IBM STARS team developed a process manual to assist software development organizations in adopting and installing the Cleanroom Engineering Software Development Process. This manual describes the process and verification activities required for performing a Cleanroom Engineering effort from the standpoint of specifiers, developers, certifiers, and managers.

The manual was developed to support process improvement programs directed at improving software quality and development productivity by incorporating some or all of the Cleanroom Engineering software development technologies into their current software development process and practices.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_299
Version: 1
Release Date: 28-FEB-91
Producer: IBM CORPORATION
Reference: CDRL 07001-001, DTIC
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: CLEANROOM ENGINEERING, SOFTWARE DEVELOPMENT, SOFTWARE PROCESS
Distribution: Approved for public release, distribution is unlimited

CLEANROOM SOFTWARE PROCESS CASE STUDY

This case study describes the problem selected for evaluating software development process capabilities for the STARS Program. This document selects a portion of the process defined in "The Cleanroom Engineering Software Development Process" (ASSET_A_299; IBM CDRL 07001-001) for use in planning the implementation of a well-defined process, in support of IBM's evaluation of the process enactment support tool, the Knowledge-based Integration Shell (KI Shell).

The "Cleanroom Software Process Case Study" problem also provides the "Host-at-Sea Buoy" problem, specified in box structure notation, for use in testing the implemented software processes.

This document is a companion document to the "Cleanroom Engineering Process Assistant Specification" (ASSET_A_274; IBM CDRL 03706-001A) and the "Cleanroom Software Process Case Study: Executive Summary" (ASSET_A_188; IBM CDRL 03706-001B).

Order Number: ASSET_A_273
Version: 1
Release Date: 26-JUL-91
Producer: IBM CORPORATION
Reference: CDRL 03706-001, DTIC AD-B157204, STARS-03706-001
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: CASE STUDY, CLEANROOM ENGINEERING, KI SHELL, SOFTWARE PROCESS, SOFTWARE PROCESS ENACTMENT SUPPORT TOOLS
Distribution: Approved for public release, distribution is unlimited

CLEANROOM SOFTWARE PROCESS CASE STUDY: EXECUTIVE SUMMARY

This document provides an executive summary of the Cleanroom Software Process Case Study, describing the case study's purpose and artifacts produced as a result of the STARS IS-15 Process task.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_188
Version: 1
Release Date: 26-JUL-91
Producer: IBM CORPORATION
Reference: CDRL 03706-001B, DTIC AD-B157379
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: CLEANROOM PROCESS
Distribution: Approved for public release, distribution is unlimited

CLEANROOM SOFTWARE PROCESS CASE STUDY: LESSONS LEARNED FROM STARS TASK IS-15

The intent of this paper is to describe the Cleanroom Software Process Case Study and the lessons learned in defining and modeling a software development process, selecting a portion of the process for an enactment experiment, and the implementation of the selected portion of the software process. The Cleanroom Software Process Case Study will also be discussed from the perspective of its implications on projects interested in process-driven development.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_292
Version: 1
Release Date: 26-JUN-92
Producer: IBM CORPORATION
Reference: CDRL 04024-001B, DTIC
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: CLEANROOM ENGINEERING, LESSONS LEARNED, PROCESS ENACTMENT, PROCESS MODELING, PROCESS SYSTEM IMPLEMENTATION, SOFTWARE PROCESS
Distribution: Approved for public release, distribution is unlimited

DISTRIBUTED COMPUTING DESIGN SYSTEM (DCDS) DATA MODEL

Analysis of DCDS Data Model. This paper looks at the DCDS data model and how well it correlates with the current procedural steps taken by Boeing during project development. Information includes a basic life cycle model, a brief history of the development behind DCDS and RDD, the basic DCDS data model, and comparisons between DCDS and Boeing's System Engineering Process WBS, the Boeing Software Standards Online System, and RDD. DCDS is found to cover a broad spectrum of the life cycle development activities. The database schema language is extensive and extendible. An overview of the DCDS data model shows that DCDS is a highly structured approach to distributed design of large software development projects. There is a tremendous amount of information contained in a project database. The wide scope and sheer size of the database for a large project has its problems. There is a need for improved man machine interface (new windowing technology), multi-user capabilities, better word processing capabilities, and more eloquent database interaction.

Order Number: ASSET_A_165
Alternate Name: ANALYSIS OF DCDS DATA MODEL
Version: 1
Release Date: 14-JAN-91
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BS20-004, CDRL 3048R, DTIC AD-B157890
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: ACQUISITION LIFE CYCLE MODEL, DATA MODEL, DCDS, DISTRIBUTED DESIGN, LIFE CYCLE, OBJECT-ORIENTED, RDD, WBS
Distribution: Approved for public release, distribution is unlimited

DISTRIBUTED COMPUTING DESIGN SYSTEM MAPPED TO AN OBJECT-ORIENTED FRAMEWORK

This document examines the mapping of the Distributed Computing Design System (DCDS) data model into a type hierarchy of an Object Oriented Framework based on CASE Interface Services (CIS) Version 1.0. Discussion covers the characteristics of Object-Oriented Framework which facilitate tool integration and the relationship between integration and mapping. Integrating DCDS into an Object-Oriented Framework involved hierarchical placement of large grained file objects and mapping of an ERA model into an object-oriented type hierarchy.

Order Number: ASSET_A_282
Alternate Name: MAPPING THE DCDS DATA MODEL TO AN OBJECT-ORIENTED FRAMEWORK
Version: 1
Release Date: 25-JAN-91
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 3049R, DTIC AD-B158017
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: ACQUISITION LIFE CYCLE MODEL, DATA MODEL, DCDS, DISTRIBUTED COMPUTING DESIGN SYSTEM
Distribution: Approved for public release, distribution is unlimited

INFORMATION OBJECT MODELING METHODOLOGY

This report describes a methodology for specifying complex systems in a short period of time. Developed by Dr. Gerald R. White of the Foxboro company, the methodology is based on Information Object Models. This report introduces the methodology, explains what an Information Object Model is, and provides guidance on developing and reviewing diagrams as part of such models. The report also discusses the brief, yet intense history of a government-run experiment using the Information Object Modeling methodology.

Information Object Modeling is a technique for developing specification models for systems. The techniques for building Information Object Models were adapted from techniques of real-time structured analysis and the Foxboro company's experience in specifying and developing real-time process control systems.

An Information Object Model (IOM) is organized to provide levels of information for different audiences, so that one document can meet the needs of different people. A mission statement is provided which describes the scope of the system. An overview of the system describes the major functional objects. Finally, each functional object is discussed in detail.

The modeling techniques for an IOM use the graphical techniques real-time structured analysis, including transformation diagrams (data flow plus control flow), state transition diagrams, and entity relationship diagrams. Transformation diagrams, however, are applied in a different manner, representing the communication of objects organized hierarchically rather than a functional decomposition of processes.

This document is available from ASSET in paper copy only.

Order Number: ASSET_A_193
Version: 1
Release Date: 21-JUN-90
Producer: IBM CORPORATION
Reference: CDRL 01200-001A, DTIC AD-A228477
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE DOCUMENTING
Keywords: METHODOLOGY, REAL-TIME, STRUCTURED ANALYSIS
Distribution: Approved for public release, distribution is unlimited

INTEGRATING DOMAIN-SPECIFIC REUSE FOR SYSTEM/SOFTWARE ENGINEERS

This course provides system and software engineers an introduction to system development with domain-specific software reuse. A domain is a set of current and future applications marked by a set of common capabilities and data. Domain-specific software reuse is the reuse of ideas, knowledge, artifacts, personnel and software components in an existing domain. The course introduces the methods necessary to integrate domain-specific software reuse concepts into current system and software development processes by emphasizing domain analysis, generic architecture development, specific architecture development, and system composition. This course is intended for use in both government and industry training. The course could be tailored for presentation at the university level.

Order Number: ASSET_A_271
Alternate Name: CARDS COURSE DESCRIPTION
Version: PRELIM
Release Date: 19-JUN-92
Producer: PARAMAX
Reference: CDRL 04102A, STARS-AC-04102A/001/00
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE REUSE
Keywords: DOMAIN-SPECIFIC, REUSE ENGINEERING
Distribution: Distribution authorized to US Government agencies and their contractors

LIFE CYCLE MODELING LESSONS LEARNED

This document presents lessons learned from experimentation with the Distributed Computing Design System (DCDS) and building repositories with Digital Equipment Corporation's CDD/Dictionary Operator tool, CDO. This experimentation leads to a discussion on object-oriented principles and analysis which sets the foundation for explanations of the Waterfall, Spiral, and Rapid Iterative models, a Framework Information Model (FIM), and a SEE Information Model (SIM). With this foundation, the SEE framework type hierarchy and the Life Cycle Data Model (LCDM) are presented. Logical structuring of PARTITIONs, COLLECTIONs, and CONTEXTs clarify a user's visibility into the SEE framework and use of the LCDM. The LCDM is exemplified further with the instantiation of objects created during the DCDS and CDO experimentation. The Appendices contain CIS, IRDS, ATIS, and the SEE framework type hierarchies, detailed explanations of logical structuring, definitions and placement into the SEE framework type hierarchy.

Order Number: ASSET_A_171
Version: 1
Release Date: 16-AUG-91

Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BS20-010, CDRL 3050R, DTIC AD-B157889
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: LESSONS LEARNED, LIFE CYCLE DATA MODEL, OBJECT-ORIENTED, RAPID ITERATIVE MODEL, SEE FRAMEWORK, SOFTWARE LIFE CYCLE, SPIRAL MODEL, WATERFALL MODEL
Distribution: Approved for public release, distribution is unlimited

POLICY MANAGEMENT EXTENSIONS LESSONS LEARNED - FINAL REPORT

This report captures the lessons learned from extending the Prototype Engineering Information System (PREIS) framework to include a process management functionality. The PREIS framework was extended to provide a solution to the International Software Process Workshop (ISPW-6) problem. This work was performed by Honeywell under subcontract to Boeing as part of the STARS Program.

This document is available from ASSET in PostScript format only, or on paper.

Order Number: ASSET_A_263
Alternate Name: FINAL REPORT LESSONS LEARNED ON POLICY MANAGEMENT EXTENSIONS
Version: 1
Release Date: 17-MAY-91
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BS25-010, CDRL 3032R, DTIC AD-B158018
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: OBJECT-ORIENTED, POLICY MANAGEMENT, PREIS, PROCESS MODEL
Distribution: Approved for public release, distribution is unlimited

POLICY MANAGEMENT MODEL EXTENSIONS

This report details the extensions that were made to the Prototype Engineering Information System (PREIS) in order to implement a solution to the International Software Process Workshop (ISPW-6) problem. The report provides the extensions of the type hierarchy that were needed to model the problem as well as the activity and policy descriptions that provide the behavioral part of the solution. This work was performed by Honeywell under subcontract to Boeing as part of the STARS program.

This document is available from ASSET in PostScript format only or on paper.

Order Number: ASSET_A_194
Version: 1
Release Date: 26-JUL-91
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BS25-011, CDRL 3030R, DTIC AD-B158020
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: OBJECT-ORIENTED, PREIS, PROCESS MODEL
Distribution: Approved for public release, distribution is unlimited

PROCESS ENACTMENT DEMONSTRATION

This document is one of a group describing features of process enactment of the Boeing CDD/Repository based environment. This document describes the type extensions made for the process example problem to enable process enactment in the CDD/Repository based environment. This document also describes the software that implements the example problem, the installation of the software, and usage scenarios to demonstrate the example problem. The process enactment mechanism used to enable this example is described in Boeing CDRL 4027R-1 (ASSET_A_302: CDD/Repository Enactment Mechanism Development)

This document is available from ASSET in PostScript format only, or on paper.

Order Number: ASSET_A_303
Alternate Name: STARS PROCESS ENACTMENT DEMONSTRATION TYPES AND SCENARIOS
Version: 1
Release Date: 24-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 4027R-2
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: CDD/REPOSITORY, CONTROL INTEGRATION, CONTROL POINTS AND POLICY, DATA INTEGRATION, ISPW-6 PROCESS EXAMPLE, PROCESS ENACTMENT, PROCESS MODELING, PROCESS SCENARIOS, SOFTWARE ENGINEERING ENVIRONMENT, STARS
Distribution: Distribution authorized to US Government agencies and their contractors

PROCESS NOTATION DEVELOPMENT: AAA NOTATION

This document is one of a group describing features of process notation for the Boeing CDD/Repository based environment. This document describes the Agents, Artifacts, and Activities (AAA) notation being developed to interface with the Control Points and Policies enactment mechanism in the CDD/Repository based environment. This document also describes the examples of processes. The process enactment mechanism used to enable this notation is described in Boeing CDRL 4027R-1 (ASSET_A_302: CDD/Repository Enactment Mechanism Development).

This document is available from ASSET in paper form only.

Order Number: ASSET_A_309
Alternate Name: STARS PROCESS NOTATION DEVELOPMENT: AAA NOTATION AAA
Version: 1
Release Date: 24-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 4024R-6
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: CDD/REPOSITORY, ISPW-6, PROCESS ENACTMENT, PROCESS MODELING, PROCESS NOTATION, SOFTWARE ENGINEERING ENVIRONMENT, STARS
Distribution: Distribution authorized to US Government agencies and their contractors

PROCESS NOTATION DEVELOPMENT: AAA TOOLING REQUIREMENTS

This document is one of a group describing features of process notation for the Boeing CDD/Repository based environment. This document describes tooling requirements for encoding processes using the Agents, Artifacts, and Activities (AAA) notation being developed to interface with the Control Points and Policies enactment mechanism in the CDD/Repository based environment. This document also describes the examples of processes. The process enactment mechanism used to enable this notation is described in Boeing CDRL 4027R-1 (ASSET_A_302: CDD/Repository Enactment Mechanism Development).

This document is available from ASSET in paper form only.

Order Number: ASSET_A_310
Alternate Name: STARS PROCESS NOTATION DEVELOPMENT: AAA TOOLING REQUIREMENTS AAA
Version: 1
Release Date: 24-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 4024R-7
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: CDD/REPOSITORY, ISPW-6, PROCESS ENACTMENT, PROCESS MODELING, SOFTWARE ENGINEERING ENVIRONMENT, STARS
Distribution: Distribution authorized to US Government agencies and their contractors

PROCESS TRANSFER EXPERIMENT RESULTS

This document describes the STARS "Process Transfer Experiment" that was conducted during the "I" increment, and presents the lessons learned and experiences from performing a process transfer. The "Process Transfer Experiment" that was performed was designed to meet two objectives: 1) To investigate aspects and characteristics of a process transfer, by performing one, and 2) To prepare an SPMS model suitable to support the planning of Cleanroom Software engineering efforts.

The goal of the "Process Transfer Experiment" was to understand the requirements for a process asset that is to be placed in a process asset library, in terms of its ability to be understood and applied by process asset library consumers. This report discusses our observations about process asset characteristics required for a process asset to be understood and applied.

This document is available from ASSET in PostScript and Word for Windows formats only, and on paper.

Order Number: ASSET_A_293
Version: FINAL
Release Date: 26-JUL-92
Producer: IBM CORPORATION
Reference: CDRL 04028-002, DTIC AD-B167545
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: CLEANROOM ENGINEERING, PROCESS ASSET LIBRARY, PROCESS MANAGEMENT, PROCESS MODEL, PROCESS REUSE, PROCESS TRANSFER, SPMS
Distribution: Approved for public release, distribution is unlimited

REUSE LIBRARY PROCESS MODEL

This delivery of the STARS Reuse Library Process Model contains the Domain Analysis Process building block. Domain analysis is defined as the process by which information used in developing software systems is identified, captured, and organized with the purpose of making it reusable. The objective of the process model is to convert the ad hoc nature of domain analysis into a repeatable procedure with well-defined, tangible outputs.

Order Number: ASSET_A_177
Version: 1
Release Date: 26-JUL-91
Producer: IBM CORPORATION
Reference: CDRL 03041-002, DTIC AD-B157091
Asset Type: DOCUMENT
Domains: REUSE LIBRARY, SOFTWARE DEVELOPMENT PROCESS
Keywords: DOMAIN ANALYSIS
Distribution: Approved for public release, distribution is unlimited

RISK REDUCTION REASONING-BASED DEVELOPMENT PARADIGM TAILORED NAVY C2 SYSTEMS

This report defines a STARS Navy Command and Control Process Model (NCCPM). The NCCPM is the result of

tailoring the STARS Composite Process Model (defined in a separate report entitled "Draft Composite Paradigm Report", STARS report number 03068) to the Navy Tactical Command and Control domain.

The NCCPM describes the entire systems development life cycle from early concept through contract award, design, development, and operations and maintenance, with an emphasis on reuse-based software development.

Order Number: ASSET_A_196
Alternate Name: NCCPM
Version: 1
Release Date: 31-JAN-91
Producer: PARAMAX
Reference: CDRL 03070, DTIC AD-B157659, PUBLICATION NO. GR-7670-1219(NP)
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE REUSE
Keywords: COMMAND AND CONTROL, DOMAIN SPECIFIC, FRAMEWORK, PROCESS MODEL, RISK REDUCTION, SCPM, SOFTWARE REUSE, TRUSTED SYSTEM
Distribution: Distribution authorized to US Government agencies and their contractors

SOFTWARE PROCESS MANAGEMENT SYSTEM STUDENT HANDOUT

This document is the student handout prepared for the SEI/STARS Asset Acquisition Sub-task training class. The student handout covers basic aspects of process management with respect to the Software Process Management System. The handout also includes several sample workshop problems.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_286
Alternate Name: SPMS TRAINING CLASS STUDENT HANDOUT
Version: 1
Release Date: 30-SEP-91
Producer: IBM CORPORATION
Reference: CDRL 03705-001B, DTIC AD-B157201
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: CLEANROOM PROCESS
Distribution: Approved for public release, distribution is unlimited

SOFTWARE PROCESS TOOLS AND TECHNIQUES EVALUATION REPORT

This document describes the tools and technology examined on STARS Task IS-15. This report summarizes the IBM team's examination of software process representation tools and techniques. It also summarizes the examination of software process enactment tools, and techniques for implementing a process system from a well-defined system of processes, such as the "Cleanroom Engineering Software Development Process".

The software process definition tools and techniques sections of the document: 1) examines the feasibility of porting the Software Process Management System (SPMS) from the Apple Macintosh to the IBM STARS SEE, 2) provides an SPMS Port plan and 3) discusses the use of box structures as a notation for recording aspects of software processes. The software process enactment tools and techniques sections of the document: 1) describes the KI Shell tool selected for supporting the IBM STARS "Cleanroom Software Process Case Study", 2) describes the specification, design and implementation of the "Cleanroom Engineering Process Assistant" prototype, and 3) provides lessons learned from performing the "Cleanroom Software Process Case Study". Finally the document makes recommendations for the selection of software process definition and enactment support capabilities for the IBM STARS SEE.

This document is available from ASSET in PostScript or SGML format, or on paper.

Order Number: ASSET_A_255
Version: 1
Release Date: 30-SEP-91
Producer: IBM CORPORATION
Reference: CDRL 03705-001, DTIC AD-A255946
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: EVALUATION, TOOLS
Distribution: Approved for public release, distribution is unlimited

SOFTWARE-FIRST LIFE CYCLE - FINAL DEFINITION

The Software-First Life Cycle (SFLC) is a dramatically different approach to systems development using rapid prototyping, software reuse, concurrent engineering, and other new technologies. By combining these into a well-defined, integrated, and highly concurrent process, the SFLC offers the potential for considerable improvements in software productivity, quality, and reusability.

This document formally defines each phase of the Software-First Life Cycle using the ETVX (Entry Criteria, Task, Validation, and Exit Criteria) model, and describes the input, output, and intermediate work products.

Order Number: ASSET_A_197
Version: 1
Release Date: 15-JAN-90
Producer: IBM CORPORATION
Reference: CDRL 01240-001, DTIC AD-A228483
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: PROCESS MODEL
Distribution: Approved for public release, distribution is unlimited

SPMS/CLEANROOM ENGINEERING SOFTWARE DEVELOPMENT PROCESS MODEL - MODEL A

The purpose of this document is to present the artifacts for the SPMS/Cleanroom Engineering Software Development Process Model - Model "A". Two SPMS models were produced from this experiment, namely the "A" model and the "B" model. The "A" model was created from an analysis of the Cleanroom Engineering Software Development Process (SDP) manual and incorporates fixes to minor errors found from the analysis of the Cleanroom Engineering SDP, as well as suggested process enhancements. The "B" model was also based on the Cleanroom Engineering SDP and incorporates fixes to minor errors as well, but did not incorporate the suggested process enhancements.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_294
Alternate Name: PROCESS TRANSFER EXPERIMENT
Version: 1
Release Date: 26-JUL-92
Producer: IBM CORPORATION
Reference: CDRL 04028-001A, DTIC
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: CLEANROOM, GENERIC PROCESS MODEL, PROCESS MODELING, SOFTWARE PROCESS MANAGEMENT
Distribution: Approved for public release, distribution is unlimited

SPMS/CLEANROOM ENGINEERING SOFTWARE DEVELOPMENT PROCESS MODEL - MODEL B

The purpose of this document is to present the artifacts for the SPMS/Cleanroom Engineering Software Development Process Model - Model "B". Two SPMS models were produced from this experiment, namely the "A" model and the "B" model. The "A" model was created from an analysis of the Cleanroom Engineering Software Development Process (SDP) manual and incorporates fixes to minor errors found from the analysis of the Cleanroom Engineering SDP, as well as suggested process enhancements. The "B" model was also based on the Cleanroom Engineering SDP and incorporates fixes to minor errors as well, but did not incorporate the suggested process enhancements.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_295
Alternate Name: PROCESS TRANSFER EXPERIMENT
Version: 1
Release Date: 26-JUL-92
Producer: IBM CORPORATION
Reference: CDRL 04028-001B, DTIC
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: CLEANROOM, GENERIC PROCESS MODEL, PROCESS MODELING, SOFTWARE PROCESS MANAGEMENT
Distribution: Approved for public release, distribution is unlimited

STARS COMPOSITE PROCESS MODEL (SCPM)

This report defines the STARS Composite Process Model (SCPM), a reuse- based paradigm for the development of trusted, high performance systems. The SCPM represents an adaptation and melding of previous DARPA, SEI, STARS, and industry process modeling work in the areas of risk-based activities, high performance, trusted system development, software reuse, library support for reuse, and trusted systems reuse issues. In particular, it provides an initial framework for the development of trusted systems, a high risk domain of growing importance, as the need for more complex systems and more systems interoperability increases.

Order Number: ASSET_A_190
Alternate Name: DRAFT COMPOSITE PARADIGM REPORT
Version: 1
Release Date: 30-MAY-91
Producer: PARAMAX
Reference: CDRL 03068, DTIC AD-B157660, PUBLICATION NO. GR-7670-1195(NP)
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: DOMAIN SPECIFIC, FRAMEWORK, PROCESS MODEL, RISK REDUCTION, SOFTWARE REUSE, TRUSTED SYSTEM
Distribution: Distribution authorized to US Government agencies and their contractors

TOOLS/NOTATION EVALUATION REPORT

This report represents a continuation of the work defined in "Software Process Tools and Techniques Report" to report on IBM's work in evaluating and employing tools to support process modeling and design. This report focuses on the use of SPMS for supporting process modeling and design, critically assesses the strengths and weaknesses of the SPMS approach, and identifies SPMS's applicability for use on DoD projects. This includes these supporting papers:

- Defining, Simulating, and Validating Software Process Models in SPMS
- Using SPMS to Support Process Modeling and Project Planning
- Lesson Learned in Prototyping and Use of a Software Process Modeling System
- SPMS/R Component Function Descriptions

This document is available from ASSET in PostScript, SGML and Word for Windows formats, and also on paper.

Order Number: ASSET_A_275
Alternate Name: PROCESS TOOLS/NOTATIONS REPORT

Version: 1
Release Date: 26-JUL-92
Producer: IBM CORPORATION
Reference: CDRL 04024-002, DTIC AD-B166804
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS
Keywords: PROCESS MODELING, PROJECT PLANNING, SOFTWARE PROCESS MANAGEMENT
Distribution: Distribution authorized to US Government agencies and their contractors

SOFTWARE DOCUMENTING

ADA PROLOGUE GENERATION TOOL

Accurate, up-to-date documentation facilitates maintenance of programs and their potential reuse on other projects. The Ada Prologue Generation Tool (ProGen) is a prototype tool developed by Boeing to automatically embed documentation into Ada source code. This documentation, which is stored in program prologues per STARS standards, includes exceptions raised, pre-defined and implementation defined pragmas, and the usage of Chapter 13 LRM language features. This information is obtained by ProGen via scanning of the program source code.

Order Number: ASSET_A_157
Alternate Name: PROGEN
Version: 1.0
Release Date: 26-MAR-89
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BQ10-039, CDRL 0260-5, DTIC AD-B157095
Asset Type: SOFTWARE CODE
Domains: SOFTWARE DOCUMENTING, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: DOCUMENTATION, PROLOG GENERATOR
Distribution: Distribution authorized to US Government agencies and their contractors

AIR TRAFFIC CONTROL INFORMATION OBJECT MODEL

Information Object Modeling is a technique for developing specification models for systems. The techniques for building Information Object Models were adapted from techniques of real-time structured analysis and the Foxboro company's experience in specifying and developing real-time process control systems.

An Information Object Model (IOM) is organized to provide levels of information for different audiences, so that one document can meet the needs of different people. A mission statement is provided which describes the scope of the system. An overview of the system describes the major functional objects. Finally, each functional object is discussed in detail.

The modeling techniques for an IOM use the graphical techniques real-time structured analysis, including transformation diagrams (data flow plus control flow), state transition diagrams, and entity relationship diagrams. Transformation diagrams, however, are applied in a different manner, representing the communication of objects organized hierarchically rather than a functional decomposition of processes.

This document is a specification model for an air traffic control system prepared using Real Time Structured Analysis. It shows Foxboro's concept of specification packaging and can serve as an alternative to MIL-STD-2167A.

This document is available from ASSET only in SGML format, or on paper.

Order Number: ASSET_A_191
Alternate Name: STARS STRUCTURE (DOD AAS IOM DOCUMENT VERSION 1.3)
Version: 1.3
Release Date: 11-MAY-90
Producer: IBM CORPORATION
Reference: CDRL 01200-001, DTIC AD-A228479
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE DOCUMENTING
Keywords: REAL-TIME, STRUCTURED ANALYSIS
Distribution: Approved for public release, distribution is unlimited

DOCUMENT REQUIREMENTS FOR THE SOFTWARE-FIRST LIFE CYCLE

The DoD-STD-2167A military standard provides a framework of activities, work products, reviews, etc. which can be tailored by contractors to the specific requirements of a project. The STARS program began this tailoring process by developing section-by-section guidelines for tailoring data item description (DID) sections of DoD-STD-2167A CDRL items for use as Software-First Life Cycle (SFLC) CDRL items, so they can be customized to the specific requirements of the SFLC and associated technologies, e.g., prototyping, reusability, Ada software design, and object-oriented design. This was possible since there is a one-to-one mapping between DoD-STD-2167A deliverable items and SFLC deliverable items.

Sample Tailoring of 2167A DIDs for Software-First Life Cycle, is the last of five evolving documents. It contains a description of a proposed set of deliverable items which will support the SFLC, a mapping of DoD-STD-2167A deliverable items to the proposed SFLC deliverable items, and tailoring guidelines on a section-by-section basis for DoD-STD-2167A deliverable items.

Order Number: ASSET_A_198
Alternate Name: SAMPLE TAILORING OF 2167A DIDS FOR SOFTWARE-FIRST LIFE CYCLE

Version: 5
Release Date: 14-AUG-90
Producer: IBM CORPORATION
Reference: CDRL 01880-001, DTIC AD-A228487
Asset Type: DOCUMENT
Domains: SOFTWARE DOCUMENTING
Keywords: DID, DOD-STD-2167A, SOFTWARE LIFE CYCLE
Distribution: Approved for public release, distribution is unlimited

INFORMATION OBJECT MODELING METHODOLOGY

This report describes a methodology for specifying complex systems in a short period of time. Developed by Dr. Gerald R. White of the Foxboro company, the methodology is based on Information Object Models. This report introduces the methodology, explains what an Information Object Model is, and provides guidance on developing and reviewing diagrams as part of such models. The report also discusses the brief, yet intense history of a government-run experiment using the Information Object Modeling methodology.

Information Object Modeling is a technique for developing specification models for systems. The techniques for building Information Object Models were adapted from techniques of real-time structured analysis and the Foxboro company's experience in specifying and developing real-time process control systems.

An Information Object Model (IOM) is organized to provide levels of information for different audiences, so that one document can meet the needs of different people. A mission statement is provided which describes the scope of the system. An overview of the system describes the major functional objects. Finally, each functional object is discussed in detail.

The modeling techniques for an IOM use the graphical techniques real-time structured analysis, including transformation diagrams (data flow plus control flow), state transition diagrams, and entity relationship diagrams. Transformation diagrams, however, are applied in a different manner, representing the communication of objects organized hierarchically rather than a functional decomposition of processes.

This document is available from ASSET in paper copy only.

Order Number: ASSET_A_193
Version: 1
Release Date: 21-JUN-90
Producer: IBM CORPORATION
Reference: CDRL 01200-001A, DTIC AD-A228477
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE DOCUMENTING
Keywords: METHODOLOGY, REAL-TIME, STRUCTURED ANALYSIS
Distribution: Approved for public release, distribution is unlimited

SOFTWARE ENGINEERING ENVIRONMENT

ACE/EIS INTEGRATION LESSONS LEARNED

This document reports the lessons learned from integrating the Ada Command Environment (ACE) into the Prototype Engineering Information System (PREIS) environment. ACE is an interactive command language environment for Ada software development where Ada is both the programming language and the command language. PREIS is an object-oriented environment targeted for computer-aided engineering (CAE). During this task, PREIS demonstrated an ability to model several ACE constructs in an object-oriented fashion. ACE on the other hand, demonstrated its extensibility by the addition of abstract data types. These abstract data types give the user a window into the PREIS environment.

The ACE/PREIS integration effort was undertaken to identify and provide insights into issues related to tool integration in an object management system. The integration effort revealed the difficulty of integrating C with Ada and demonstrated again the necessity of good documentation. The requirement for a programmatic interface to PREIS for future integration efforts was also identified.

Order Number: ASSET_A_152
Version: 1
Release Date: 02-NOV-90
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BS20-002, CDRL 03041R, DTIC AD-B158007
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: ACE, EIS, FRAMEWORK, LESSONS LEARNED, SOFTWARE ENGINEERING ENVIRONMENT, TOOL INTEGRATION
Distribution: Approved for public release, distribution is unlimited

ADA COMMAND ENVIRONMENT

The Ada Command Environment (ACE) is an interactive Ada environment coupled with a set of Ada abstract data types (ADTs). The interactive environment allows users to rapidly prototype general Ada applications, while the ADTs allow prototyping of applications for particular domains, such as X Window System applications. In addition, the ADTs provide an Ada view of underlying applications, which when combined with the interactive environment replaces the traditional role of a command language. When using ACE, Ada becomes the command language as well as the programming language. This version of ACE includes support for X Window System prototyping. This asset includes the ACE Users Manual.

Order Number: ASSET_A_153
Alternate Name: ACE
Version: 8.0
Release Date: 29-Oct-1990
Producer: PARAMAX
Reference: CDRL 00990, DTIC AD-A228822, GR-7670-1141 (NP)
Asset Type: SOFTWARE CODE
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: COMMAND ENVIRONMENT, OBJECT-ORIENTED, PROGRAMMING-IN-THE-LARGE
Distribution: Approved for public release, distribution is unlimited

ADA COMMAND ENVIRONMENT USERS MANUAL

This technical report provides users of the Ada Command Environment (ACE) with a description of each of the packages and subprograms that are provided with ACE. In addition, the report describes the overall purpose of ACE, general guidelines on ACE usage, and provides examples of typical user interaction during an ACE session.

Order Number: ASSET_A_278
Alternate Name: ACE USERS MANUAL
Version: 8.0
Release Date: 29-OCT-90
Producer: PARAMAX
Reference: CDRL 00980, DTIC AD-A229400, STARS-RC-00980/001/00
Asset Type: SOFTWARE DOCUMENTATION
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: COMMAND ENVIRONMENT, OBJECT-ORIENTED, PROGRAMMING-IN-THE-LARGE
Distribution: Approved for public release, distribution is unlimited

ADA FORMAL METHODS IN THE STARS ENVIRONMENT

This report is a collection of several possible ways in which tools supporting formal methods might be made interoperable and/or integrated into the STARS SEE. The possibilities discussed are merely representative, and are limited only by the effort available for completing the report. In general, formal methods might be applied to any phase of system development, including requirements analysis, design, implementation of both hardware and software, testing, and so forth. However, this report is restricted for the most part to formal code verification of Ada. (The single exception to this restriction is the discussion of run-time checking in Anna.) Formal code verification is proof that an algorithm implemented by a piece of code meets a formally stated specification. ORA's Penelope and Ariel tools support code verification

Order Number: ASSET_A_256
Version: 1
Release Date: 03-JUN-92
Producer: PARAMAX
Reference: CDRL 03089, STARS-SC-03089/001/00
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: FORMAL METHODS, SOFTWARE DEVELOPMENT TOOLS, STARS SEE, TOOL INTEGRATION
Distribution: Distribution authorized to US Government agencies and their contractors

ADA INTERFACES TO POSIX

This report considers the applicability of the Portable Operating System Interface (POSIX) to the development of the Software Engineering Environment (SEE) for the Software Technology for Adaptable, Reliable Systems (STARS) program. It compares and contrasts characteristics and potential overlap of object management interfaces (in this example, CAIS-A) and POSIX. Because there are overlaps in the objectives for POSIX and an object manager, there are some apparent overlaps in the functions provided by the interfaces. It is concluded that there is no actual overlap in function in the I/O model for persistent data nor in the execution control models. Interfaces like CAIS-A should be the portable interface set for tools, while POSIX offers a method for gaining portability for the CAIS and other interfaces in the base level of the SEE.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_155
Alternate Name: INTERFACE STANDARDS INFORMAL TECHNICAL DATA, ADA INTERFACES TO POSIX
Version: 1
Release Date: 14-APR-89
Producer: PARAMAX
Reference: CDRL 02021-005, DTIC AD-A228820
Asset Type: DOCUMENT
Domains: OPERATING SYSTEMS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: COMPARISON REPORT, INTERFACE STANDARDS, POSIX
Distribution: Approved for public release, distribution is unlimited

ADA PCTE BINDING (ADAPCTE)

This is the Paramax Ada binding to PCTE. The specification of the Ada binding is based on the ECMA (European Computer Manufacturers Association) Ada PCTE specification (Standard ECMA-162 Ada Language Binding, December 1991). The binding uses the GIE Emeraude PCTE 1.5 version 12.2 C libraries. Some changes to the ECMA specifications were made to accommodate the binding

to PCTE 1.5. The release includes the AdaPCTE Version Description Document, which describes this version of AdaPCTE and provides installation instructions. These documents are provided in both plain ASCII and PostScript forms.

Order Number: ASSET_A_257
Alternate Name: ADAPCTE
Version: 0.1
Release Date: 12-JUN-92
Producer: PARAMAX
Reference: CDRL 04014, STARS-TC-04014/001/00
Asset Type: SOFTWARE CODE
Domains: ADA STANDARDS AND BINDINGS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: BINDINGS, PCTE, PORTABLE COMMON TOOL ENVIRONMENT
Distribution: Distribution authorized to US Government agencies and their contractors

ADA PROGRAM FLOW ANALYSIS TOOL (APFAT)

APFAT is an object-based, menu-driven program which produces the following reports: main indented call tree summary, subprogram invocations, subprogram references, exceptions usage summary, exceptions raised-by summary, identifier usage summary, user specified identifier usage summary, Booch diagrams, and program metrics. These reports may be directed to the user's screen or redirected to any output file.

The program is designed to process syntactically correct Ada source code in the correct compilation order. It runs on an Apollo system with UNIX under BSD4.2 and the Verdix Compiler SRS 3.5.1.

Order Number: ASSET_A_199
Alternate Name: APFAT
Version: 1
Release Date: 17-MAR-89
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BQ10-032, CDRL 0260-2, DTIC AD-B157893
Asset Type: SOFTWARE CODE
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: BOOCH DIAGRAM, CALLING TREE, METRICS
Distribution: Approved for public release, distribution is unlimited

ADA PROLOGUE GENERATION TOOL

Accurate, up-to-date documentation facilitates maintenance of programs and their potential reuse on other projects. The Ada Prologue Generation Tool (ProGen) is a prototype tool developed by Boeing to automatically embed documentation into Ada source code. This documentation, which is stored in program prologues per STARS standards, includes exceptions raised, pre-defined and implementation defined pragmas, and the usage of Chapter 13 LRM language features. This information is obtained by ProGen via scanning of the program source code.

Order Number: ASSET_A_157
Alternate Name: PROGEN
Version: 1.0
Release Date: 26-MAR-89
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BQ10-039, CDRL 0260-5, DTIC AD-B157095
Asset Type: SOFTWARE CODE
Domains: SOFTWARE DOCUMENTING, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: DOCUMENTATION, PROLOG GENERATOR
Distribution: Distribution authorized to US Government agencies and their contractors

ADA SEMANTIC INTERFACE SPECIFICATION (ASIS)

The Ada Semantic Interface Specification is a layered vendor independent open architecture. ASIS queries and services provide a consistent interface to information within the Ada Program Library. Clients of ASIS are shielded and free from the implementation details of each Ada vendor's proprietary library and intermediate representations.

This document consists solely of Ada package (design) specifications with no accompanying software or other documentation.

Order Number: ASSET_A_313
Version: 0.4
Release Date: 21-OCT-91
Producer: PARAMAX
Reference: CDRL 251901-003, DTIC
Asset Type: SOFTWARE DOCUMENTATION
Domains: ADA STANDARDS AND BINDINGS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: ADA COMPILER, ADA PROGRAM LIBRARY, ADA SEMANTIC INFORMATION, ASIS, INTERFACE, OPEN ARCHITECTURE, VENDOR-INDEPENDENT
Distribution: Approved for public release, distribution is unlimited

ADA STATIC STRUCTURE EVALUATION TOOL (ASSET)

The Ada Static Structure Evaluation Tool (ASSET) is a port and enhancement of the SIMTEL 20 tool, "Compilation Order", (COMPORD). ASSET reads in Ada source code from a user-supplied list of files and writes dependency and compilation order reports either to standard output or to a user-specified file. ASSET can serve as a front end to another STARS tool, the Ada Program Flow Analysis Tool

(APFAT)

Order Number ASSET_A_202
Alternate Name ASSET
Version 2.0
Release Date 17-MAR-89
Producer BOEING DEFENSE AND SPACE GROUP
Reference Boeing Accession No. BQ10-34, CDRL 0260-3, DTIC AD-B157894
Asset Type SOFTWARE CODE
Domains SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords ADA, COMPILATION ORDER
Distribution Approved for public release, distribution is unlimited

AFS LESSONS LEARNED ON STARS - UPDATED

Along with other STARS contractors, the Boeing STARS program is engaged as a DARPA Beta site of the Andrew File System (AFS). This document contains the lessons learned from Boeing's experience with the Andrew File System, on the STARS Project

Order Number ASSET_A_189
Alternate Name UPDATED AFS LESSONS LEARNED ON STARS
Version 2
Release Date 01-AUG-91
Producer BOEING DEFENSE AND SPACE GROUP
Reference Boeing Accession No. BS20-007, CDRL 3056R, DTIC AD-B157857
Asset Type DOCUMENT
Domains SOFTWARE ENGINEERING ENVIRONMENT, STARS PROGRAM MANAGEMENT
Keywords AFS, FILE SYSTEM, INTERNET, LESSONS LEARNED, NETWORK, STARS, WIDE AREA NETWORK
Distribution Approved for public release, distribution is unlimited

ANALYSIS OF EXISTING SOFTWARE ENGINEERING ENVIRONMENTS

Presents the results of an analysis of four key Ada software engineering environments -- the Ada Language System / Navy (ALS/N), the Rational R1000 Development System, the Software Development and Maintenance Environment (SDME) and the Space Station Software Support Environment (SSE). The study recommends that the STARS Software Engineering Environment incorporate such capabilities as software process management, standard interfaces, data base support, and software engineering support, including support for Ada, multi-target development, prototyping, and reuse

Order Number ASSET_A_163
Alternate Name: ENVIRONMENT CAPABILITY MATRIX
Version: 1
Release Date: 17-MAR-89
Producer: IBM CORPORATION
Reference: CDRL 00110-001, DTIC AD-A228485
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: ENVIRONMENT
Distribution: Approved for public release, distribution is unlimited

CDD/REPOSITORY ENACTMENT MECHANISM DEVELOPMENT

This document is one of a group describing features of process enactment of the Boeing CDD/Repository based environment. This document describes the specification, design, and type extensions made for the process mechanism to enable process enactment in the CDD/Repository based environment.

This document also describes the software that implements the mechanism, the installation and testing of the software, and usage of the mechanism. A demonstration example using the process enactment mechanism to enable process enactment is described in Boeing CDRL 4027R-2, (ASSET_A_303: Process Enactment Demonstration).

This document is available from ASSET in PostScript format only, or on paper.

Order Number: ASSET_A_302
Alternate Name: STARS ENACTMENT MECHANISM DEVELOPMENT ON CDD/REPOSITORY
Version: 1
Release Date: 24-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 4027R-1
Asset Type: DOCUMENT
Domains: REUSE LIBRARY, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: COHESION, CONTROL INTEGRATION, DATA INTEGRATION, PROCESS ENACTMENT, PROCESS MODELING, SOFTWARE ENGINEERING ENVIRONMENT, STARS
Distribution: Distribution authorized to US Government agencies and their contractors

HYPERTEXT APPLICATION INFORMAL REPORT

This report describes the results of work done at the Boeing Defense and Space Group (D&SG) STARS Program to investigate and develop an application, using a commercial hypertext tool, to support reusable asset understanding and peer review.

Order Number: ASSET_A_300

Version: 1
Release Date: 21-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 1850
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: HYPERTEXT, LESSONS LEARNED, REUSABLE ASSET, SGML, STARS, TAG
Distribution: Distribution authorized to US Government agencies and their contractors

INTER-TOOL COMMUNICATION FACILITY (ITCF)

The Inter Tool Communication Facility (ITCF) is a VAX/CAIS-A facility. It is designed to allow concurrently executing tools to cooperate in an integrated manner while maintaining a high degree of modularity and functional independence. The ITCF offers a layered architecture of communication services which may be adapted to meet the requirements of a variety of tool integration strategies. The principle conceptual layers are streams, stream protocols, and messages.

The ITCF services manage the complex details of inter-tool communication and provide conceptually simple and easy-to-use interfaces to the tool writer. These services facilitate the integration of tools with relatively modest effort. This includes existing tools which were originally not designed to use the ITCF packages.

This document presents an overview of ITCF concepts together with a review of design problems and solutions. A discussion of selected tools is presented and a description of their adaptation is provided. So that users may demonstrate the facility, instructions are provided for running the tools within the environment. Finally, lessons learned are documented with conclusions as to how they can be applied to future enhancements and modifications.

Order Number: ASSET_A_167
Alternate Name: INTER-TOOL COMMUNICATION FACILITY (ITCF) FINAL REPORT
Version: 1
Release Date: 13-FEB-90
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BR00-490, CDRL 1270, DTIC AD-A240479
Asset Type: SOFTWARE DOCUMENTATION
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: ARCHITECTURE, COMMUNICATION SERVICES, LESSONS LEARNED, TOOL INTEGRATION
Distribution: Approved for public release, distribution is unlimited

IRIS/DIANA ANALYSIS

This technical report analyzes the requirements of the STARS environment from the perspective of intermediate language use and programmatic access to compiler information. The report details several strategies for intermediate language use in an environment for the purpose of tight integration of tool suites. In addition a detailed analysis of the currently available STARS intermediate languages is presented. These include DIANA (Descriptive Intermediate Attributed Notation for Ada), IRIS (Intermediate Representation Including Semantics), and ASIS (Ada Semantics Interface Specification). The detailed analysis compares each intermediate language with the purposes with which it was designed and the overall goals of the STARS environment. The report concludes with a recommendation to standardize on a hybrid - IRIS as the intermediate language and ASIS as a programmatic interface. It also provides a strategy to accomplish the standardization. An appendix with examples of intermediate language use is included.

Order Number: ASSET_A_169
Alternate Name: STARS INTERMEDIATE LANGUAGE ASSESSMENT, IRIS/DIANA ANALYSIS
Version: 1
Release Date: 30-JUL-90
Producer: PARAMAX
Reference: CDRL 01430, DTIC AD-A229349
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: DIANA, INTERMEDIATE LANGUAGE, STANDARDS
Distribution: Approved for public release, distribution is unlimited

LIFE CYCLE MODELING LESSONS LEARNED

This document presents lessons learned from experimentation with the Distributed Computing Design System (DCDS) and building repositories with Digital Equipment Corporation's CDD/Dictionary Operator tool, CDO. This experimentation leads to a discussion on object-oriented principles and analysis which sets the foundation for explanations of the Waterfall, Spiral, and Rapid Iterative models, a Framework Information Model (FIM), and a SEE Information Model (SIM). With this foundation, the SEE framework type hierarchy and the Life Cycle Data Model (LCDM) are presented. Logical structuring of PARTITIONs, COLLECTIONs, and CONTEXTs clarify a user's visibility into the SEE framework and use of the LCDM. The LCDM is exemplified further with the instantiation of objects created during the DCDS and CDO experimentation. The Appendices contain CIS, IRDS, ATIS, and the SEE framework type hierarchies, detailed explanations of logical structuring, definitions and placement into the SEE framework type hierarchy.

Order Number: ASSET_A_171
Version: 1
Release Date: 16-AUG-91

Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BS20-010, CDRL 3050R, DTIC AD-B157889
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: LESSONS LEARNED, LIFE CYCLE DATA MODEL, OBJECT-ORIENTED, RAPID ITERATIVE MODEL, SEE FRAMEWORK, SOFTWARE LIFE CYCLE, SPIRAL MODEL, WATERFALL MODEL
Distribution: Approved for public release, distribution is unlimited

OBJECT MANAGER AND FRAMEWORK SPECIFICATION COMPARISONS

A framework provides the facilities necessary to build a Software Engineering Environment (SEE). These include a repository to store the project data, tool and data integration facilities, and facilities to define and control the engineering process. There are currently a number of existing and proposed frameworks. This document compares four specifications for framework services (A Tool Integration Standard, the Common Ada Programming Support Environment Interface Set, the Portable Common Tool Environment, and Engineering Information Services) in the areas of process and policy management, database capabilities, and tool integration.

Order Number: ASSET_A_173
Version: 1
Release Date: 04-JAN-91
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BS20-004, CDRL 3029R, DTIC AD-B157887
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: TOOL
Distribution: Approved for public release, distribution is unlimited

PCTE BROWSER TOOL

The PCTE Browser Tool (PBT) is designed to graphically display parts of a PCTE object base. Selected objects in the object base and the relationships among these objects are displayed at the PBT user's request. The PBT is intended to complement text-oriented commands that are included with the Emeraude PCTE 1.5 release. PBT version 0.1 is an alpha release of the browser.

The PBT is an instance of the Reusable Graphical Browser (RGB), a generic graphical browser for the display of networks of nodes and arcs. In the case of the PBT, the nodes displayed by the RGB are PCTE objects, and the arcs are PCTE links. PBT version 0.1 was developed using RGB version 0.5. The PBT is an X Window System application, and requires the installation of X11. It has been built and tested using Release 4 of X11; however, it is expected that it also should be usable under X11R3 or X11R5. It was developed using the Paramax STARS Ada implementation of Ada/Xt and Ada implementation of some MIT Athena and Hewlett Packard widgets, version 3.3.

The PBT is ultimately intended for use in an ECMA PCTE environment, and has been implemented using the ECMA-162 Ada programming bindings to PCTE. However, in the absence of a conforming ECMA PCTE implementation, the PBT has been built on top of the Emeraude V12.2 PCTE implementation, using the subset implementation of the ECMA Ada binding developed by Paramax STARS (version 0.1).

Order Number: ASSET_A_258
Alternate Name: PBT
Version: 0.1
Release Date: 12-JUN-92
Producer: PARAMAX
Reference: CDRL 04014, STARS-TC-04014/002/00
Asset Type: SOFTWARE CODE
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: BROWSER, GRAPHICS, PCTE, PORTABLE COMMON TOOL ENVIRONMENT
Distribution: Distribution authorized to US Government agencies and their contractors

PCTE BROWSER TOOL USER MANUAL

The PCTE Browser Tool (PBT) is an instance of the Paramax STARS Reusable Graphical Browser (RGB), a generic graphical browser for the display of networks of nodes and arcs. In the case of the PBT, the nodes displayed by the RGB are PCTE objects, and the arcs are PCTE links. The PBT is an X Window System application, and as such makes use of the Paramax STARS implementation of Ada/Xt and Ada implementation of some MIT Athena and Hewlett Packard widgets.

This document describes the general capabilities of the PBT and how to invoke them. For detailed instructions on how to install or rebuild the PBT, consult the associated Version Description Document (VDD).

Order Number: ASSET_A_259
Alternate Name: PBT USER MANUAL
Version: 0.1
Release Date: 12-JUN-92
Producer: PARAMAX
Reference: CDRL 04014, STARS-TC-04014/003/00
Asset Type: SOFTWARE DOCUMENTATION
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: BROWSER, GRAPHICS, PCTE, PORTABLE COMMON TOOL ENVIRONMENT, USER MANUAL
Distribution: Distribution authorized to US Government agencies and their contractors

PROCESS NOTATION DEVELOPMENT: AAA NOTATION

This document is one of a group describing features of process notation for the Boeing CDD/Repository based environment. This document describes the Agents, Artifacts, and Activities (AAA) notation being developed to interface with the Control Points and Policies enactment mechanism in the CDD/Repository based environment. This document also describes the examples of processes. The process enactment mechanism used to enable this notation is described in Boeing CDRL 4027R-1 (ASSET_A_302: CDD/Repository Enactment Mechanism Development).

This document is available from ASSET in paper form only.

Order Number: ASSET_A_309
Alternate Name: STARS PROCESS NOTATION DEVELOPMENT: AAA NOTATION AAA
Version: 1
Release Date: 24-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 4024R-6
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: CDD/REPOSITORY, ISPW-6, PROCESS ENACTMENT, PROCESS MODELING, PROCESS NOTATION, SOFTWARE ENGINEERING ENVIRONMENT, STARS
Distribution: Distribution authorized to US Government agencies and their contractors

PROCESS NOTATION DEVELOPMENT: AAA TOOLING REQUIREMENTS

This document is one of a group describing features of process notation for the Boeing CDD/Repository based environment. This document describes tooling requirements for encoding processes using the Agents, Artifacts, and Activities (AAA) notation being developed to interface with the Control Points and Policies enactment mechanism in the CDD/Repository based environment. This document also describes the examples of processes. The process enactment mechanism used to enable this notation is described in Boeing CDRL 4027R-1 (ASSET_A_302: CDD/Repository Enactment Mechanism Development).

This document is available from ASSET in paper form only.

Order Number: ASSET_A_310
Alternate Name: STARS PROCESS NOTATION DEVELOPMENT: AAA TOOLING REQUIREMENTS AAA
Version: 1
Release Date: 24-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 4024R-7
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: CDD/REPOSITORY, ISPW-6, PROCESS ENACTMENT, PROCESS MODELING, SOFTWARE ENGINEERING ENVIRONMENT, STARS
Distribution: Distribution authorized to US Government agencies and their contractors

PROTOTYPE SEE LESSONS LEARNED

This document describes the lessons learned from tailoring an ATIS (A Tool Integration Standard) based object-oriented framework for use as the Boeing STARS (Software Technology for Adaptable Reliable Systems) environment. An information model was constructed based on an analysis of the Boeing STARS project. This model served as the basis for tailoring the framework and for presenting the analysis to the eventual users of the environment. The objects identified in the information model and their interrelationships were the foundation for extensions to the framework. During the work, an iterative process evolved for extending an object-oriented framework.

The iterative process, the information model, the framework extensions, and the rationale for the choices are presented. Lessons learned in developing the information model and the user interface design are discussed.

Order Number: ASSET_A_176
Version: 1
Release Date: 15-AUG-91
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BS20-012, CDRL 3056R, DTIC AD-B158019
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: DEVELOPMENT ENVIRONMENT, LESSONS LEARNED, OBJECT-ORIENTED
Distribution: Distribution authorized to US Government agencies and their contractors

REUSABLE GRAPHICAL BROWSER (RGB)

This reusable software component implements a graphical user interface for browsing over the contents of an object management system. It is intended to facilitate the construction of various browsing tools by serving as a user interface component for those tools. The generic application interface and tailorable user interface provided by this component allow it to be adapted to a wide range of browsing applications. They also serve to promote the portability of graphical browsing tools by insulating them from the details of the underlying graphics system. Because the details of the graphics system are hidden by an abstract application interface, tool builders need not be familiar with the intricacies of graphics packages or window systems in order to use this product. The RGB is implemented in Ada, with the exception of one small routine coded in C. The current implementation uses the MIT X Toolkit: Ada Language Implementation (based on X Version 11, Release 3) as the underlying graphics system. This makes it compatible with numerous

hardware platforms.

Order Number: ASSET_A_183
Alternate Name: RGB
Version: 0.5
Release Date: 16-JAN-92
Producer: PARAMAX
Reference: CDRL 03714, STARS-TC-03714/005/00
Asset Type: SOFTWARE CODE
Domains: SOFTWARE ENGINEERING ENVIRONMENT, USER INTERFACE
Keywords: GRAPHICAL BROWSER, GRAPHICS, OBJECT MANAGEMENT SYSTEM, USER INTERFACE
Distribution: Approved for public release, distribution is unlimited

REUSABLE GRAPHICAL BROWSER (RGB) USERS MANUAL

This document is the User's Manual for the Reusable Graphical Browser developed by the Paramax STARS Team. The Reusable Graphical Browser (also referred to as the Reusable Browser) is a reusable software component designed to facilitate the construction of graphical tools for browsing over the contents of various object management systems. More specifically, it is intended to serve as a basis for constructing such tools quickly and easily using the Ada programming language. The purpose of this Manual is to provide guidance to tool builders using the Reusable Graphical Browser to construct specific graphical browser tools.

The User's Manual presents both a conceptual description of the Reusable Graphical Browser and detailed instructions for its reuse. Among other information, it contains complete technical specifications for the application (browser tool) interface provided by the Reusable Graphical Browser and a general description of the user (man-machine) interface implemented by the Reusable Graphical Browser. It does not, however, contain instructions regarding the use of tools constructed from the Reusable Graphical Browser. Such instructions are necessarily application-specific.

Order Number: ASSET_A_277
Alternate Name: RGB USERS MANUAL
Version: 0.5
Release Date: 31-JAN-92
Producer: PARAMAX
Reference: CDRL 03715, STARS-TC-03715/004/00
Asset Type: SOFTWARE DOCUMENTATION
Domains: SOFTWARE ENGINEERING ENVIRONMENT, USER INTERFACE
Keywords: GRAPHICAL BROWSER, OBJECT MANAGEMENT SYSTEMS
Distribution: Approved for public release, distribution is unlimited

ROAMS TESTBED REPORT AND LESSONS LEARNED

The Boeing Defense & Space Group (DS&G) STARS Program designed and developed the Reusable Object Access and Management System (ROAMS), and integrated it with the system engineering environment (SEE) being developed by D&SG for the STARS program. ROAMS is a reuse library mechanism for an object-oriented, repository based software reuse library. This document describes the results and lessons learned during the ROAMS development and integration, and during ROAMS demonstrations with the SEE. Included are discussions of the ROAMS design history, major issues encountered and their resolutions, design issues on logical and physical reuse libraries, SEE environment features with navigators, element (object) editors, and the ROAMS extensions to the type hierarchy.

Order Number: ASSET_A_301
Alternate Name: REPORT ON RESULTS AND LESSONS LEARNED FROM ROAMS TESTBED OPERATIONS
Version: 1
Release Date: 22-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 1870
Asset Type: DOCUMENT
Domains: REUSE LIBRARY, SOFTWARE ENGINEERING ENVIRONMENT
Keywords: ASSET, ATIS, LIBRARIAN, OBJECT-ORIENTED, RETRIEVE, REUSE LIBRARY, ROAMS, SLCSE, STARS, TAXONOMY
Distribution: Distribution authorized to US Government agencies and their contractors

SEE DEMONSTRATION REPORT

This document describes an all-day demonstration of CASE tools as applied to software engineering environments, which was held at IBM's Gaithersburg facility on July 10, 1991. There were 5 presentations, and 12 demonstrations of 17 products. This report explains what was demonstrated, describes how the demonstration was conducted, identifies the participants and observers, and presents conclusions and recommendations resulting from the demonstration. Included are copies of the foils that were presented as well as handout material from the demonstration.

Order Number: ASSET_A_179
Version: 1
Release Date: 26-JUL-91
Producer: IBM CORPORATION
Reference: CDRL 03034-001, DTIC AD-B157267
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: DEMONSTRATION
Distribution: Approved for public release, distribution is unlimited

SEE IMPLEMENTATION PLAN

The Boeing STARS SEE (Software Engineering Environment) is an integrated environment providing support for activities across the entire engineering life cycle. This plan identifies the process we are using to develop the SEE and it shows a schedule for development of the requisite functionality.

This document is available from ASSET in PostScript format only, which is missing three figures, or on paper.

Order Number: ASSET_A_304
Version: 1
Release Date: 31-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 4035R
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: SOFTWARE ENGINEERING ENVIRONMENT, STARS
Distribution: Distribution authorized to US Government agencies and their contractors

SEE INFORMATION MODEL

This document describes the Information Model of the current Boeing STARS systems engineering environment. The information model is described in the object-oriented style of ATIS (A Tool Integration Standard). The current model supports project planning, Ada software development, and the ROAMS reuse library.

Order Number: ASSET_A_311
Version: 1
Release Date: 31-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 4034R
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: ATIS, INFORMATION MODEL, SOFTWARE ENGINEERING ENVIRONMENT, STARS
Distribution: Distribution authorized to US Government agencies and their contractors

SEE TECHNICAL REPORT

This document describes STARS work in the area of software engineering environments, including a description of the initial instantiation of an IBM SEE solution, experiences in the use of the capabilities of the initial SEE solution on real projects, and discussions of new and emerging technologies being developed and evaluated by the IBM STARS team for inclusion into the SEE.

Order Number: ASSET_A_182
Version: 1
Release Date: 26-JUL-91
Producer: IBM CORPORATION
Reference: CDRL 03032-001, DTIC AD-B157222
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Distribution: Approved for public release, distribution is unlimited

SEE/FRAMEWORK INTEGRATION LESSONS LEARNED: ALPHA TEST PROJECT

This document describes the IBM STARS Alpha SEE (Software Engineering Environment) solutions used by the Alpha Test Projects and the lessons learned from their use during the T-Increment time period (August '91 to July '92). The previously documented lessons learned from the S-Increment time period (August '90 to July '91) are included in the appendix. The lessons learned from the S-Increment reflect, for the most part, the frontend efforts to understand the tools/methods and some early experiences in the use of them. The lessons learned from the T-Increment reflect, for the most part, later experiences in the use of the specific tools/methods and the introduction of new ones.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_297
Version: 1
Release Date: 26-JUL-92
Producer: IBM CORPORATION
Reference: CDRL 04036-001A, DTIC
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: AIX CASE, FRAMEWORK INTEGRATION, LESSONS LEARNED, SOFTWARE ENGINEERING ENVIRONMENT, STARS
Distribution: Approved for public release, distribution is unlimited

SEE/FRAMEWORK INTEGRATION LESSONS LEARNED: ENTERPRISE II

This report provides an overview of the Enterprise II Software Engineering Environment and comments on its use. The comments were derived from experiences in a four day Enterprise II user/tool integrator class and a case study to define an environment for a small project. The evaluation was part of the STARS Software Engineering Environment Integration task.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_298
Version: 1
Release Date: 26-JUL-92
Producer: IBM CORPORATION
Reference: CDRL 04036-001B, DTIC AD-B167597
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT

Keywords: ENTERPRISE II, FRAMEWORK INTEGRATION, LESSONS LEARNED, PCTE, SOFTWARE ENGINEERING ENVIRONMENT, STARS, SYSECA
Distribution: Approved for public release, distribution is unlimited

STARS PROTOTYPE SEE TOOL INTEGRATION LESSONS LEARNED

A framework provides the facilities necessary to build a Software Engineering Environment (SEE). These include a repository to store the project data, tool and data integration facilities, and facilities to define and control the engineering process.

The purpose of this task was to integrate a sample tool or set of tools into a selected framework and report on the experiences and the knowledge gained from the integration effort. The framework used for this study provides an object-oriented interface that is an extension of the ATIS (A Tool Integration Standard) standard. The report begins with an overview of the tools and the framework, then describes the integration method, and concludes with the evaluation of the framework and the lessons learned.

Order Number: ASSET_A_187
Version: 1
Release Date: 01-AUG-91
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BS20-011, CDRL 3044R, DTIC AD-B158014
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: FRAMEWORK, LESSONS LEARNED, OBJECT-ORIENTED, TOOL INTEGRATION
Distribution: Approved for public release, distribution is unlimited

STARS SEE ALPHA TEST PLAN

This document describes the approach for performing the "alpha test" of the initial STARS SEE solution, defined by IBM. The approach taken involves the selection of DoD projects which will use the capabilities of the initial SEE solution, designated as the IBM STARS Alpha SEE, in one or more Ada software projects.

Order Number: ASSET_A_288
Alternate Name: SEE ALPHA TEST PLAN
Version: 1
Release Date: 28-MAR-91
Producer: IBM CORPORATION
Reference: CDRL 03029-001, DTIC
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: ALPHA TEST, STARS SEE
Distribution: Distribution authorized to US Government agencies and their contractors

TOOLS/NOTATION EVALUATION REPORT: EXPERIMENT DEFINITION TOOL REPORT

This report introduces a process modeling experiment. This experiment surveyed, selected, and used commercial modeling tools to determine whether they were appropriate tool candidates for composing process components into an integrated process model for a life-cycle process. This report introduces the experiment, identifies the International Software Process Workshop (ISPW-6) process modeling example as the problem to model, and provides a summary of conclusions about requirements for tooling determined by using the Lessons Learned report and the Process Model Documentation for each of the tools evaluated. The initial set of documents describes using International Software Systems, Inc.'s Proto+ as a modeling tool for process. This initial set is delivered as CDRL 4024R-2 (ASSET_A_307: Tools/Notation Evaluation Report: Proto Evaluation) and CDRL 4024R-3 (ASSET_A_308: Tools/Notation Evaluation Report: Proto Process Modeling). A second set of documents will be produced for Ascent Logic's Requirements Driven Development tool, RDD, when the process modeling experiment is completed. These will be delivered as CDRL 4024R-4 and CDRL 4024R-5.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_306
Version: 0.1
Release Date: 24-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 4024R-1
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: CONTROL INTEGRATION, DATA INTEGRATION, ISPW-6, LESSONS LEARNED, PROCESS ENACTMENT, PROCESS MODEL, SOFTWARE ENGINEERING ENVIRONMENT, STARS
Distribution: Distribution authorized to US Government agencies and their contractors

TOOLS/NOTATION EVALUATION REPORT: PROTO EVALUATION

This report is the second volume in a series documenting an experiment to develop a process model for a selected example process using available commercial modeling tools. This volume documents the lessons learned and evaluation of International Software Systems, Inc.'s Proto+ and PProto functional modeling tools to model an example process. The experiment is documented in CDRL 4024R-1 (ASSET_A_306: Tools/Notation Evaluation Report: Experiment Definition Tool Report). The experiment surveyed, selected, and used commercial modeling tools to determine whether they were appropriate tool candidates for composing process components into an integrated process

model for a life-cycle process. The Sixth International Software Process Workshop (ISPW-6) process modeling example was selected as the problem to model. In addition to the problem definition and summaries provided in the first volume of this experiment, the details and lessons learned developing this Proto model are documented in this volume. The third volume, CDRL 4024R-3 (ASSET_A_308: Tools/Notation Evaluation Report: Proto Process Modeling Proto Model) documents the process model.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_307
Version: 0.1
Release Date: 24-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 4024R-2
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: ISPW-6, LESSONS LEARNED, PROCESS MODEL, PROCESS SIMULATION, PROTO+, SOFTWARE LIFE-CYCLE, STARS
Distribution: Distribution authorized to US Government agencies and their contractors

TOOLS/NOTATION EVALUATION REPORT: PROTO PROCESS MODELING

This report is the third volume in a series documenting an experiment to develop a process model for a selected example process using available commercial modeling tools. This volume documents the process model developed using International Software Systems, Inc.'s Proto+ and PProto functional modeling tools to model an example process. The experiment is documented in CDRL 4024R-1 (ASSET_A_306: Tools/Notation Evaluation Report: Experiment Definition). The experiment surveyed, selected, and used commercial modeling tools to determine whether they were appropriate tool candidates for composing process components into an integrated process model for a life-cycle process. The Sixth International Software Process Workshop (ISPW-6) process modeling example was selected as the problem to model. In addition to the problem definition and summaries provided in the first volume of this experiment, the details and lessons learned developing this Proto model are documented in the second volume, CDRL 4024R-2 (ASSET_A_307: Tools/Notation Evaluation Report: Proto Evaluation).

This document is available from ASSET in paper form only.

Order Number: ASSET_A_308
Version: 0.1
Release Date: 24-JUL-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 4024R-3
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT
Keywords: ISPW-6, PROCESS MODEL, PROCESS SIMULATION, PROTO+, STARS
Distribution: Distribution authorized to US Government agencies and their contractors

SOFTWARE REUSE

ACQUISITION HANDBOOK - CARDS

The Acquisition Handbook is aimed towards Government Program Managers and their support personnel, such as Contracting Officers and Administrators, legal support, and program control, involved in systems, subsystems and component acquisition. This guidebook will assist them in incorporating software reuse into all phases of the acquisition life cycle, from concept exploration to Post Deployment Software Support (PDSS). The goal of the Acquisition Handbook is to encourage software reuse during the acquisition and maintenance portions of the life cycle process, ranging from planning the acquisition strategy through awarding the contract to managing the effort and follow-on support.

Software reuse guidance will be presented by providing methods, examples, recommendations and techniques to implement various reuse strategies throughout the acquisition life cycle. The implications and affects of software reuse on the technical, management, cost, schedule, and risk aspects of a program/system during the acquisition process will be the foundation of this document.

This document is available from ASSET in PostScript format, or on paper.

Order Number: ASSET_A_276
Alternate Name: CARDS ACQUISITION HANDBOOK
Version: DRAFT
Release Date: 04-SEP-92
Producer: PARAMAX
Reference: CDRL 04105, DTIC
Asset Type: DOCUMENT
Domains: SOFTWARE REUSE
Keywords: ACQUISITION LIFE CYCLE, SOFTWARE MAINTENANCE
Distribution: Distribution authorized to US Government agencies and their contractors

APPLICATION BLUEPRINTS

Application blueprints serve as a framework for designing new systems in an application domain, leading to reuse of design information and greater reuse of code. This document defines the term application blueprint, tells how to create one, and discusses the benefits and drawbacks of this approach. The appendix presents a generic specification and information about the initial domain analysis for creating an application blueprint for an air traffic control system. This paper can be the basis for future research on reusing analysis and design information.

Order Number: ASSET_A_186
Alternate Name: UPDATED APPLICATION BLUEPRINT DEFINITION FOR C3
Version: 1
Release Date: 10-JAN-90
Producer: IBM CORPORATION
Reference: CDRL 01490-001A, DTIC AD-A228471
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE REUSE
Keywords: APPLICATION BLUEPRINTS, DOMAIN ANALYSIS
Distribution: Approved for public release, distribution is unlimited

CARDS TRAINING PLAN

This Training Plan is designed to function as a guide for conducting domain-specific software reuse training. The Plan addresses training four different audiences. Guidelines are provided on conducting training and developing training materials for Department of Defense (DoD) Organizations, DoD Contractors, System/Software Engineers, and University Professors. The four courses may be implemented either individually or as an integrated package. Feedback and suggestions are welcomed.

Order Number: ASSET_A_272
Alternate Name: TRAINING PLAN - CENTRAL ARCHIVE FOR REUSABLE DEFENSE SOFTWARE
Version: 00A
Release Date: 07-JUL-92
Producer: PARAMAX
Reference: CDRL 04101, STARS-AC-04101/001/00A
Asset Type: DOCUMENT
Domains: SOFTWARE REUSE
Keywords: DOMAIN-SPECIFIC
Distribution: Distribution authorized to US Government agencies and their contractors

CONSOLIDATED REUSE LESSONS LEARNED

This document is a digest of lessons learned with respect to reuse or in performing work related to reuse. The document consolidates reuse lessons learned across the STARS program with lessons being contributed by each of the STARS primes (Boeing, IBM, Paramax - A Unisys Company) and by the national reuse library ASSET. Since the document is arranged as a digest, each major section may be read independently to review lessons in one area. The lessons learned address topics such as cooperative work, reuse supporting technology, and technical transition.

Order Number: ASSET_A_305
Version: 1
Release Date: 31-JAN-92
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 04049RA, DTIC
Asset Type: DOCUMENT
Domains: SOFTWARE REUSE
Keywords: COOPERATIVE WORK, DOMAIN ANALYSIS, LESSONS LEARNED, REUSE, STARS, WIDE-AREA NETWORK FILE SYSTEM
Distribution: Distribution authorized to US Government agencies and their contractors

EVALUATION OF REUSABLE SOFTWARE ENGINEERING ASSETS

Prepared by Software Engineering Technology, Inc.

This report formulates criteria and procedures for the evaluation of reusable software engineering assets. The task is set in the context of the ASSET reuse library which has the mission of helping to stimulate a national industry based on software reuse. As such, the facility must become a national authority on the subject of reuse, an advocate for the industry, and a self-sustaining business operation.

The document defines four levels of quality: 1) Documented, 2) Audited, 3) Validated, and 4) Certified. It also divides evaluation of assets into a hierarchical taxonomy, with inheritance. It then proceeds to identify the criteria by which different types of assets must be evaluated in order to meet each level of quality. The process of evaluation is defined and a sample evaluation is included.

Order Number: ASSET_A_252
Alternate Name: CRITERIA AND IMPLEMENTATION PROCEDURES FOR EVALUATION OF REUSABLE ASSETS
Version: 1
Release Date: 28-MAR-92
Producer: IBM CORPORATION
Reference: CDRL 04014-002B, DTIC AD-B166803
Asset Type: DOCUMENT
Domains: SOFTWARE REUSE
Keywords: EVALUATION CRITERIA, QUALITY ASSURANCE, SOFTWARE ENGINEERING, SOFTWARE REUSE
Distribution: Approved for public release, distribution is unlimited

FEDERAL ACQUISITION REGULATION AND BUDGET/FINANCE IMPACTS ON SOFTWARE REUSE

The Federal Acquisition Regulation (FAR) and DoD FAR Supplement (DFARS), as well as service and agency supplements to the FAR, have been examined for impediments in the way data rights are acquired and software is contracted. The current FAR environment with regard to software development, software reusability, and the use of commercially available software has been documented in this report. Initial proposed changes to the FAR and proposed FAR language changes have also been incorporated.

This document is available from ASSET only in PostScript format, and on paper.

Order Number: ASSET_A_312
Alternate Name: REUSABLE SOFTWARE ACQUISITION CURRENT FAR AND BUDGET/FINANCE ENVIRONMENTS
Version: 1
Release Date: 30-MAR-91
Producer: PARAMAX
Reference: CDRL 03501/03504, DTIC
Asset Type: DOCUMENT
Domains: SOFTWARE REUSE
Keywords: BUDGET/FINANCE, DFAR, FAR, IMPEDIMENTS TO REUSE, INCENTIVES FOR REUSE, SOFTWARE ACQUISITION, SOFTWARE REUSE
Distribution: Distribution authorized to US Government agencies and their contractors

INTEGRATING DOMAIN-SPECIFIC REUSE FOR SYSTEM/SOFTWARE ENGINEERS

This course provides system and software engineers an introduction to system development with domain-specific software reuse. A domain is a set of current and future applications marked by a set of common capabilities and data. Domain-specific software reuse is the reuse of ideas, knowledge, artifacts, personnel and software components in an existing domain. The course introduces the methods necessary to integrate domain-specific software reuse concepts into current system and software development processes by emphasizing domain analysis, generic architecture development, specific architecture development, and system composition. This course is intended for use in both government and industry training. The course could be tailored for presentation at the university level.

Order Number: ASSET_A_271
Alternate Name: CARDS COURSE DESCRIPTION
Version: PRELIM
Release Date: 19-JUN-92
Producer: PARAMAX
Reference: CDRL 04102A, STARS-AC-04102A/001/00
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE REUSE
Keywords: DOMAIN-SPECIFIC, REUSE ENGINEERING
Distribution: Distribution authorized to US Government agencies and their contractors

REUSABILITY GUIDELINES

This technical report proposes guidelines for the design and coding of reusable software parts. The proposed guidelines are intended for use by the STARS Prime Contractors and Subcontractors to foster the development of software components that offer common capabilities within well-defined application domains and that are suitable for installation in a library of reusable parts. The requirements presented by the guidelines are directed principally to software developers and, secondarily, to maintainers of a reusable parts library. Each proposed guideline or requirement is accompanied by a rationale, and where possible, illustrative examples. The guidelines are based upon STARS task Q9 experience and interpretations of the materials listed in the references section of the report.

This document is available from ASSET in ASCII text form or on paper.

Order Number: ASSET_A_180
Alternate Name: BASELINE ADA LIBRARY TECHNICAL REPORT, REUSABILITY GUIDELINES
Version: 1
Release Date: 05-MAY-89
Producer: PARAMAX
Reference: CDRL 00340-001, DTIC AD-A228819
Asset Type: DOCUMENT
Domains: SOFTWARE REUSE
Keywords: GUIDELINES
Distribution: Approved for public release, distribution is unlimited

RISK REDUCTION REASONING-BASED DEVELOPMENT PARADIGM TAILORED NAVY C2 SYSTEMS

This report defines a STARS Navy Command and Control Process Model (NCCPM). The NCCPM is the result of tailoring the STARS Composite Process Model (defined in a separate report entitled "Draft Composite Paradigm Report", STARS report number 03068) to the Navy Tactical Command and Control domain.

The NCCPM describes the entire systems development life cycle from early concept through contract award, design, development, and operations and maintenance, with an emphasis on reuse-based software development.

Order Number: ASSET_A_196
Alternate Name: NCCPM
Version: 1
Release Date: 31-JAN-91
Producer: PARAMAX
Reference: CDRL 03070, DTIC AD-B157659, PUBLICATION NO. GR-7670-1219(NP)
Asset Type: DOCUMENT
Domains: SOFTWARE DEVELOPMENT PROCESS, SOFTWARE REUSE
Keywords: COMMAND AND CONTROL, DOMAIN SPECIFIC, FRAMEWORK, PROCESS MODEL, RISK REDUCTION, SCPM, SOFTWARE REUSE, TRUSTED SYSTEM
Distribution: Distribution authorized to US Government agencies and their contractors

STANDARDS AND GUIDELINES FOR REPOSITORY DELIVERABLES

This technical report contains recommendations for guidelines and standards to be used in developing Ada programs and technical documents for delivery to a repository. It provides a proposal for standard prologues for Ada programs which are SGML-processable. A sample

SGML DTD is provided that will validate an Ada prologue coded to this standard.

An overview of SGML tools is provided together with a discussion of processing graphics integrated with text.

Order Number: ASSET_A_185
Version: 1
Release Date: 17-MAR-89
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 0320, DTIC AD-A240478
Asset Type: DOCUMENT
Domains: SOFTWARE REUSE, STARS PROGRAM MANAGEMENT
Keywords: GUIDELINES, REPOSITORY, STYLE GUIDE
Distribution: Approved for public release, distribution is unlimited

STARS REUSABILITY GUIDELINES

A set of Ada coding guidelines for component development that emphasize reusability. Code that follows these guidelines will be easier to reuse on multiple projects and platforms. Many examples are provided illustrating the guidelines.

Order Number: ASSET_A_209
Alternate Name: REPOSITORY GUIDELINES
Version: 1
Release Date: 30-APR-90
Producer: IBM CORPORATION
Reference: CDRL 01550-001, DTIC AD-A228468
Asset Type: DOCUMENT
Domains: SOFTWARE REUSE
Keywords: CODING GUIDELINES, REUSABILITY
Distribution: Approved for public release, distribution is unlimited

STARS REUSE CONCEPTS, VOLUME I - CONCEPTUAL FRAMEWORK FOR REUSE PROCESSES

This volume introduces basic STARS reuse concepts and provides a high-level definition of the STARS Conceptual Framework for Reuse Processes. Volume II will provide a more detailed definition of the Framework and the processes it encompasses. Volume III will provide a "Practitioner's View" of STARS reuse concepts, illustrating through sample scenarios how the Framework and processes can be employed in practice.

The purpose of this document is to articulate STARS concepts and expectations for reuse in the context of system and software development. This purpose is accomplished by:

- elaborating on the STARS reuse vision;
- stating STARS goals for reuse;
- defining a conceptual framework for considering and defining reuse processes;
- identifying low level reuse processes that STARS may provide as process building blocks (precise, composable process definitions) in the context of the reuse process framework and specific life cycle models;
- establishing a common terminology for reuse;
- addressing the impact and opportunities for use of distributed, heterogeneous asset libraries as a reuse-enabling technology; and,
- providing a context for understanding STARS reuse plans and products.

Order Number: ASSET_A_253
Alternate Name: CONCEPTUAL FRAMEWORK FOR REUSE PROCESSES
Version: 1.0
Release Date: 14-FEB-92
Producer: PARAMAX
Reference: CDRL 04040, DTIC AD-A247267
Asset Type: DOCUMENT
Domains: SOFTWARE REUSE
Keywords: CONCEPTUAL FRAMEWORK, REUSE PROCESS, SOFTWARE DEVELOPMENT, STARS
Distribution: Approved for public release, distribution is unlimited

STARS PROGRAM MANAGEMENT

AFS LESSONS LEARNED ON STARS - UPDATED

Along with other STARS contractors, the Boeing STARS program is engaged as a DARPA Beta site of the Andrew File System (AFS). This document contains the lessons learned from Boeing's experience with the Andrew File System, on the STARS Project.

Order Number: ASSET_A_189
Alternate Name: UPDATED AFS LESSONS LEARNED ON STARS
Version: 2
Release Date: 01-AUG-91
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BS20-007, CDRL 3056R, DTIC AD-B157857
Asset Type: DOCUMENT
Domains: SOFTWARE ENGINEERING ENVIRONMENT, STARS PROGRAM MANAGEMENT
Keywords: AFS, FILE SYSTEM, INTERNET, LESSONS LEARNED, NETWORK, STARS, WIDE AREA NETWORK
Distribution: Approved for public release, distribution is unlimited

CONCEPT OF OPERATIONS FOR USE OF AFS ACROSS STARS

The Concept of Operations for Use of AFS Across STARS (Conops) describes the use of AFS across the Software Technology For Adaptable, Reliable Systems (STARS) program. This description includes the STARS AFS network configuration, joint cell structures, file space conventions, directory structure, and an overview of planned experimentation. The adaptation of AFS for STARS work is addressed to provide support for the development of experiments for various STARS tasks and to aid in the general use of AFS for STARS efforts.

Order Number: ASSET_A_164
Version: 1
Release Date: 15-AUG-91
Producer: PARAMAX
Reference: CDRL 03015, DTIC AD-A240109
Asset Type: DOCUMENT
Domains: STARS PROGRAM MANAGEMENT
Keywords: AFS
Distribution: Approved for public release, distribution is unlimited

PRACTICAL ASPECTS OF REPOSITORY OPERATIONS

This document describes the operation of the STARS Repository computer and the basis for the policies under which it operated. The STARS Repository acted as a focal point of the STARS program and supported many STARS activities. It stored all deliverable software and technical reports. It was also a means of communication amongst the prime contractors, the subcontractors, the consultants, and the contracting agency. Electronic mail was interchanged, other electronic documents were shared, software was developed, and software was shared and reused by the STARS contract participants using the STARS Repository computer.

Order Number: ASSET_A_207
Version: 1
Release Date: 10-JAN-90
Producer: IBM CORPORATION
Reference: CDRL 01440-001, DTIC AD-A228455
Asset Type: DOCUMENT
Domains: REUSE LIBRARY, STARS PROGRAM MANAGEMENT
Keywords: GUIDEBOOK, OPERATIONS
Distribution: Approved for public release, distribution is unlimited

STANDARDS AND GUIDELINES FOR REPOSITORY DELIVERABLES

This technical report contains recommendations for guidelines and standards to be used in developing Ada programs and technical documents for delivery to a repository. It provides a proposal for standard prologues for Ada programs which are SGML-processable. A sample SGML DTD is provided that will validate an Ada prologue coded to this standard.

An overview of SGML tools is provided together with a discussion of processing graphics integrated with text.

Order Number: ASSET_A_185
Version: 1
Release Date: 17-MAR-89
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: CDRL 0320, DTIC AD-A240478
Asset Type: DOCUMENT
Domains: SOFTWARE REUSE, STARS PROGRAM MANAGEMENT
Keywords: GUIDELINES, REPOSITORY, STYLE GUIDE
Distribution: Approved for public release, distribution is unlimited

STARS 91 PROCEEDINGS

The goal of the STARS program is to increase productivity, reliability, and quality of DoD application software. STARS is approaching this by synergistically integrating support for modern software development processes and modern reuse concepts within state-of-the-art software engineering environment technology. STARS is focused on accelerating a change in the way software is developed within the DoD. This change represents a shift to a megaprogramming paradigm.

The conference program has been designed to introduce the concepts of megaprogramming and describe the role STARS is playing in the transition to this new paradigm. The plenary session provided a high level overview as well as some economic analysis of the potential benefits of megaprogramming. The closing discussion by Dr. Barry Boehm describes the relationship of megaprogramming and STARS to the DoD Software technology plan. Three of the four track sessions focused on the major elements of megaprogramming - process driven development, domain-specific reuse, and technology support. The fourth track concentrated on the STARS technology transition activities associated reuse, and technology support. The fourth track concentrated on the STARS technology transition activities associated with accelerating the shift to this new way of doing business.

This document is available from ASSET in paper form only.

Order Number: ASSET_A_254
Version: 91
Release Date: 04-DEC-91
Producer: STARS TECHNOLOGY CENTER
Reference: CDRL none, DTIC AD-A247031
Asset Type: DOCUMENT
Domains: STARS PROGRAM MANAGEMENT
Keywords: STARS 91
Distribution: Approved for public release, distribution is unlimited

TEXT PROCESSING

PEER REVIEW CAPABILITY DESCRIPTION

This document describes the prototype peer review editor that was developed as part of Boeing's STARS repository integration activities. The prototype peer review editor adapted a public domain structure editor to support the peer review of Ada source code. The editor includes a program to translate Ada source in ASCII text to an intermediate form used by the editor. The editor supports embedding peer comments in Ada source in a form that is acceptable to an Ada compiler and processable by a SGML parser.

Besides describing the Ada source code peer review capability, this document contains recommendations for use on a repository and for productization.

Order Number: ASSET_A_205
Version: 1
Release Date: 02-FEB-90
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BR00-447, CDRL 0850, DTIC AD-A240475
Asset Type: SOFTWARE DOCUMENTATION
Domains: TEXT PROCESSING
Keywords: PEER REVIEW, STRUCTURE EDITOR
Distribution: Distribution authorized to US Government agencies and their contractors

SGML DTD CREATION

SGML (Standard Generalized Markup Language) is an international standard for representing the elements and structure of electronically stored text. SGML uses Document Type Definitions (DTDs) to unify the structure of various kinds of documents. This report introduces SGML and examines the DTDs used on the STARS Program.

Order Number: ASSET_A_200
Version: 1
Release Date: 23-JAN-90
Producer: IBM CORPORATION
Reference: CDRL 01810-001, DTIC AD-A228475
Asset Type: DOCUMENT
Domains: TEXT PROCESSING
Keywords: SGML
Distribution: Approved for public release, distribution is unlimited

SGML DTDS AND USERS GUIDES FOR STARS

SGML (Standard Generalized Markup Language) is an international standard for representing the elements and structure of electronically stored text. SGML uses Document Type Definitions (DTDs) to unify the structure of various kinds of documents. This asset includes 3 DTDs used in the STARS program:

- The Briefing DTD, a new DTD for the formatting of briefing charts
- The Report DTD, a DTD that has been used for some reports in the STARS program
- A proposed STARS DTD, a DTD suggested for general usage for all future STARS deliveries.

Also included are a User's Guide detailing how to prepare documents using the Briefing DTDs and making recommendations on style.

Order Number: ASSET_A_204
Alternate Name: BRIEFING DTD USERS GUIDE
Version: 1
Release Date: 01-JUL-90
Producer: IBM CORPORATION
Reference: CDRL 01820-001, DTIC AD-A228473
Asset Type: DOCUMENT
Domains: TEXT PROCESSING
Keywords: DTD, SGML
Distribution: Approved for public release, distribution is unlimited

SGML LESSONS LEARNED

SGML (Standard Generalized Markup Language) is an international standard for representing the elements and structure of electronically stored text. SGML uses Document Type Definitions (DTDs) to unify the structure of various kinds of documents. This report summarizes the experience of building a new DTD.

Order Number: ASSET_A_203
Version: 1
Release Date: 12-JUL-90
Producer: IBM CORPORATION
Reference: CDRL 01835-001, DTIC AD-A228478
Asset Type: DOCUMENT
Domains: TEXT PROCESSING
Keywords: LESSONS LEARNED, SGML
Distribution: Approved for public release, distribution is unlimited

SGML PRODUCT REVIEW

SGML (Standard Generalized Markup Language) is an international standard for representing the elements and structure of electronically stored text. SGML uses Document Type Definitions (DTDs) to unify the structure of various kinds of documents. This report reviews commercial SGML products.

Order Number: ASSET_A_201
Version: 1
Release Date: 1-JUL-90
Producer: IBM CORPORATION
Reference: CDRL 01800-001, DTIC AD-A228476
Asset Type: DOCUMENT
Domains: TEXT PROCESSING
Keywords: SGML

Distribution: Approved for public release, distribution is unlimited

USER INTERFACE

ADA/X TOOLKIT (SOFTWARE)

This software package provides an Ada programmatic interface to a set of reusable user interface abstractions known as widgets. The software provides the full functionality of the M.I.T X Consortium Version 11 Release 4 X Window System. The software consists of three components: an Ada binding to the Xlib layer, an Ada implementation of the Xt Intrinsics layer, and an Ada widget library. The Ada binding to the Xlib layer is an upgrade of the STARS Foundations Ada Xlib binding, and provides a protocol interface including a set of graphics drawing primitives. The Xt Intrinsics layer provides a policy-free mechanism for creating and managing user interface objects. User interface objects are contained in the widget library which consists of a small set of commonly used user interface abstractions, such as scroll bars and command buttons.

Order Number: ASSET_A_162
Alternate Name: ADA/X WIDGET LIBRARY
Version: 3.3
Release Date: 16-JAN-92
Producer: PARAMAX
Reference: CDRL 03716, DTIC AD-A229637
Asset Type: SOFTWARE CODE
Domains: USER INTERFACE
Keywords: WIDGETS, X WINDOWS
Distribution: Approved for public release, distribution is unlimited

ADA/X TOOLKIT ARCHITECTURE

This report provides a detailed description of the Ada/Xt toolkit architecture. The purpose of this report is to describe the Ada/Xt architecture in terms of system-independent package specifications, and to describe the analysis which contributed to major design decisions. The emphasis on system-independent package specifications rather than language independent specifications derives from recognition that the C language interfaces defined in the X Toolkit (Xt) Intrinsics definition are nearly sufficiently language independent -- for languages in the Algol tradition (including Ada). The Ada toolkit design verifies this claim, since there is a very close syntactic mapping of types and interfaces from the Ada specification to the C specification.

Order Number: ASSET_A_160
Alternate Name: ADA/XT ARCHITECTURE: DESIGN REPORT
Version: 1.1
Release Date: 25-JAN-90
Producer: PARAMAX
Reference: CDRL 01000, DTIC AD-A228827
Asset Type: SOFTWARE DOCUMENTATION
Domains: USER INTERFACE
Keywords: ARCHITECTURE
Distribution: Approved for public release, distribution is unlimited

ADA/X TOOLKIT INTERFACE STYLE GUIDE

This technical report defines conventions for development of effective application user interfaces. The guide is intended for use by application developers and user interface developers using the Paramax STARS Ada/X User Interface Software. The intent is to produce a consistent style among Ada applications. The Style Guide addresses three areas of style:

- application/user interactions or dialogues,
- conventions for developing applications, and
- formal and informal user interface standardization

Order Number: ASSET_A_161
Alternate Name: ADA/X INTERFACE STYLE GUIDE
Version: 1
Release Date: 20-NOV-90
Producer: PARAMAX
Reference: CDRL 01030, DTIC AD-A229350
Asset Type: SOFTWARE DOCUMENTATION
Domains: USER INTERFACE
Keywords: STYLE GUIDE, TOOLKIT
Distribution: Approved for public release, distribution is unlimited

REUSABLE GRAPHICAL BROWSER (RGB)

This reusable software component implements a graphical user interface for browsing over the contents of an object management system. It is intended to facilitate the construction of various browsing tools by serving as a user interface component for those tools. The generic application interface and tailorable user interface provided by this component allow it to be adapted to a wide range of browsing applications. They also serve to promote the portability of graphical browsing tools by insulating them from the details of the underlying graphics system. Because the details of the graphics system are hidden by an abstract application interface, tool builders need not be familiar with the intricacies of graphics packages or window systems in order to use this product. The RGB is implemented in Ada, with the exception of one small routine coded in C. The current implementation uses the MIT X Toolkit: Ada Language Implementation (based on X Version 11, Release 3) as the underlying graphics system. This makes it compatible with numerous hardware platforms.

Order Number: ASSET_A_183
Alternate Name: RGB
Version: 0.5

Release Date 16 JAN 92
Producer PARAMAX
Reference CDRL 03714, STARS-TC-03714-005/00
Asset Type SOFTWARE CODE
Domains SOFTWARE ENGINEERING ENVIRONMENT, USER INTERFACE
Keywords GRAPHICAL BROWSER, GRAPHICS, OBJECT MANAGEMENT SYSTEM, USER INTERFACE
Distribution Approved for public release, distribution is unlimited

REUSABLE GRAPHICAL BROWSER (RGB) USERS MANUAL

This document is the User's Manual for the Reusable Graphical Browser developed by the Paramax STARS Team. The Reusable Graphical Browser (also referred to as the Reusable Browser) is a reusable software component designed to facilitate the construction of graphical tools for browsing over the contents of various object management systems. More specifically, it is intended to serve as a basis for constructing such tools quickly and easily using the Ada programming language. The purpose of this Manual is to provide guidance to tool builders using the Reusable Graphical Browser to construct specific graphical browser tools.

The User's Manual presents both a conceptual description of the Reusable Graphical Browser and detailed instructions for its reuse. Among other information, it contains complete technical specifications for the application (browser tool) interface provided by the Reusable Graphical Browser and a general description of the user (man-machine) interface implemented by the Reusable Graphical Browser. It does not, however, contain instructions regarding the use of tools constructed from the Reusable Graphical Browser. Such instructions are necessarily application-specific.

Order Number ASSET_A_277
Alternate Name RGB USERS MANUAL
Version 0.5
Release Date 31-JAN-92
Producer PARAMAX
Reference CDRL 03715, STARS-TC-03715/004/00
Asset Type SOFTWARE DOCUMENTATION
Domains SOFTWARE ENGINEERING ENVIRONMENT, USER INTERFACE
Keywords GRAPHICAL BROWSER, OBJECT MANAGEMENT SYSTEMS
Distribution Approved for public release, distribution is unlimited

SERPENT LESSONS LEARNED ON STARS

An application program can be divided into the application code itself, which does the work of the application, and its user interface code, which links the application code with the user. A User Interface Management System (UIMS) is a tool that is used to design a user interface separately from an application program so that one may be changed without affecting the other. This document examines the Serpent User Interface Management System to determine if this type of user interface generator is useful in prototyping and developing user interfaces for STARS tools. The intended audience for this document are project managers and user interface developers.

Order Number: ASSET_A_184
Version: 1
Release Date: 30-NOV-90
Producer: BOEING DEFENSE AND SPACE GROUP
Reference: Boeing Accession No. BS20-003, CDRL 3055R, DTIC AD-B157856
Asset Type DOCUMENT
Domains: USER INTERFACE
Keywords: LESSONS LEARNED, SERPENT, UIMS, USER INTERFACE MANAGEMENT SYSTEM
Distribution Distribution authorized to US Government agencies and their contractors

USER INTERFACE MANAGEMENT SYSTEMS

This report examines three user interface management systems (UIMS) for their suitability as a UIMS for the STARS Software Engineering Environment. There are many different definitions of the UIMS, but the core of the definition that a UIMS should separate the management of the user interface, both the representation and dialogue aspects, from the application semantics. Most UIMS include some type of graphical layout capability, usually either an editor or layout language. Dialogue languages for handling user interaction is an area of active research.

The three UIMSs chosen for evaluation were Serpent, developed by Software Engineering Institute (SEI), Chiron-1, under development in the Arcadia project, and TeleUse, a product marketed by TeleLogic (U.S. Subsidiary of TeleSoft). The three UIMSs chosen for this study represent the range of sources for UIMS technology.

Order Number: ASSET_A_192
Alternate Name: PROCESS ENVIRONMENT USER INTERFACE MANAGEMENT SYSTEM
Version: 1
Release Date: 31-OCT-90
Producer: PARAMAX
Reference: CDRL 01100, DTIC AD-B152499, PUBLICATION NO. GR-7670-1156(NP)
Asset Type: DOCUMENT
Domains: USER INTERFACE
Keywords: CHIRON-1, COMPARISON REPORT, SERPENT, TELEUSE, UIMS
Distribution: Approved for public release, distribution is unlimited

USER META INTERFACE

This document reports the results of the Boeing STARS task BR24, User Meta Interface, (also called the STARS User Interface Toolkit or SUITE). It includes the status of the prototype work done on BR24 and a comparison code size based on the prototype sample application. There is a

technical discussion of the implementation of SUITE and recommendations for future work are identified.

This document is available from ASSET in TEXT format (minus the figures), and on paper.

Order Number:	ASSET_A_195
Alternate Name:	FINAL REPORT ON BOEING TASK R24, A USER META INTERFACE
Version:	1
Release Date:	01-JUL-90
Producer:	BOEING DEFENSE AND SPACE GROUP
Reference:	CDRL 1000, DTIC AD-A240474
Asset Type:	DOCUMENT
Domains:	USER INTERFACE
Keywords:	SUITE, USER INTERACTION TASKS, USER INTERFACE, VIRTUAL INTERFACE, X WINDOWS
Distribution:	Approved for public release, distribution is unlimited

INDEX

ACE/EIS Integration Lessons Learned 11, 22
Acquisition Handbook - CARDS 32
Ada Command Environment Users Manual 11, 23
Ada Command Environment 11, 22
Ada Formal Methods In The STARS Environment 12, 23
Ada Interfaces To POSIX 6, 23
Ada PCTE Binding (ADAPCTE) 3, 23
Ada Program Flow Analysis Tool (AP-FAT) 12, 24
Ada Prologue Generation Tool 21, 24
Ada Report Production System (ARPS) - Technical Report 4
Ada Report Production System (ARPS) 4
Ada Semantic Interface Specification (ASIS) 3, 24
Ada Static Structure Evaluation Tool (ASSET) 12, 24
Ada/SQL Bindings: General Definition Of Project 4, 5
Ada/SQL Bindings: Users Manual 4, 5
Ada/SQL Bindings 3, 5
Ada/X Toolkit (software) 38
Ada/X Toolkit Architecture 38
Ada/X Toolkit Interface Style Guide 38
AFS Lessons Learned On STARS - Updated 25, 35
Air Traffic Control Information Object Model 12, 21
AMS (Asset Management System) Product Definition Document 6
Analysis Of Existing Software Engineering Environments 25
Application Blueprints 13, 32
ASSET Catalog 6
Asset Library Open Architecture Framework (ALOAF) 7
CARDS Training Plan 33
CDD/Repository Enactment Mechanism Development 7, 25
CEPA Development Activity Log 13
Cleanroom Engineering Document Abstracts: STARS Task IS-15 13
Cleanroom Engineering Process Assistant (CEPA) Specifications 13
Cleanroom Engineering Requirements For Planning Support 14
Cleanroom Engineering Software Development Process 14
Cleanroom Software Process Case Study: Executive Summary 15
Cleanroom Software Process Case Study: Lessons Learned From STARS Task IS-15 15
Cleanroom Software Process Case Study 14
Command Center Domain Model Description - CARDS 7
Command Center Library - Technical Concept 8
Concept Of Operations For Use Of AFS Across STARS 36
Consolidated Reuse Lessons Learned 33
Distributed Computing Design System (DCDS) Data Model 15
Distributed Computing Design System Mapped To An Object-oriented Framework 15
Document Requirements For The Software-first Life Cycle 21
Evaluation Of Reusable Software Engineering Assets 33
Federal Acquisition Regulation And Budget/Finance Impacts On Software Reuse 33
Hypertext Application Informal Report 25
Information Object Modeling Methodology 16,22
Integrating Domain-specific Reuse For System/Software Engineers 16, 34
Inter-tool Communication Facility (ITCF) 26
Iris/Diana Analysis 26
Life Cycle Modeling Lessons Learned 16, 26

MACH Vs. UNIX Performance Report 6
Object Manager And Framework Specification Comparisons 27
PCTE Browser Tool User Manual 27
PCTE Browser Tool 27
Peer Review Capability Description 37
Policy Management Extensions Lessons Learned - Final Report 17
Policy Management Model Extensions 17
Practical Aspects Of Repository Operations 8, 36
Process Enactment Demonstration 17
Process Notation Development: AAA Notation 17, 28
Process Notation Development: AAA Tooling Requirements 18, 28
Process Transfer Experiment Results 18
Prototype SEE Lessons Learned 28
Repository Guidelines And Standards 8
Repository Operations And Procedures 8
Repository Specifications 8
Reusability Guidelines 34
Reusability Library Framework 9
Reusable Graphical Browser (RGB) Users Manual 29, 39
Reusable Graphical Browser (RGB) 28, 38
Reuse Library Process Model 9, 18
Reuse Library Prototype - Librarians Guide 9
Reuse Library Prototype 9
Risk Reduction Reasoning-based Development Paradigm Tailored Navy C2 Systems 18, 34
RLF AdaKnet Users Manual 9
RLF AdaTAU Users Manual 10
RLF Graphical Browser Users Manual 10
RLF Librarian Users Manual 10
Roams Testbed Report And Lessons Learned 10, 29
SEE Demonstration Report 29
SEE Implementation Plan 30
SEE Information Model 30
SEE Technical Report 30
SEE/Framework Integration Lessons Learned: Alpha Test Project 30
SEE/Framework Integration Lessons Learned: Enterprise II 30
Serpent Lessons Learned On STARS 39
SGML DTD Creation 37
SGML DTDS And Users Guides For Stars 37
SGML Lessons Learned 37
SGML Product Review 37
Software Process Management System Student Handout 19
Software Process Tools And Techniques Evaluation Report 19
Software-first Life Cycle - Final Definition 19
SPMS/Cleanroom Engineering Software Development Process Model - Model A 20
SPMS/Cleanroom Engineering Software Development Process Model - Model B 20
Standards And Guidelines For Repository Deliverables 34, 36
STARS 91 Proceedings 36
STARS Composite Process Model (SCPM) 20
STARS Prototype See Tool Integration Lessons Learned 31
STARS Reusability Guidelines 35
STARS Reuse Concepts, Volume I - Conceptual Framework For Reuse Processes 35
STARS SEE Alpha Test Plan 31

Tools/Notation Evaluation Report: Experiment Definition Tool Report....31
Tools/Notation Evaluation Report: Proto Evaluation....31
Tools/Notation Evaluation Report: Proto Process Modeling....32
Tools/Notation Evaluation Report....20
User Interface Management Systems....39
User Meta Interface....39

ASSET

Asset Source for Software Engineering Technology

User Account Application Form

This form is used to request an account on the STARS ASSET Reuse Library.
Send the completed and signed form to:

ASSET Account Manager
2611 Cranberry Square
Morgantown, WV 26505

Phone: (304) 594-1762
FAX: (304) 594-3951

[For ASSET use only]

Account _____

Date _____

Signature _____

A. Personal information.

Name _____
(include title or rank) First MI Last

Organization _____

Mail Address _____

_____

City _____ State _____ Zip _____ Country _____

Phone (_____) _____ FAX (_____) _____

Electronic Address _____

B. Computer/terminal information.

If you do not need a computer account, check here and skip to section C _____

Point of Contact Name _____
(Person who manages the computer/terminal used.)

Point of Contact Phone (_____) _____

Communications method used to access ASSET (check all that apply):

___Internet ___Direct Dial (Baud rate)_____

Communications Software _____

Terminal location (check one):

___US/Govt. Building ___Commercial ___Campus ___Other_____

[Please complete both sides of this form.]

ASSET User Account Application Form -- Side 2

C. User Category Information.

These questions pertain to the work and/or application for which you will be accessing the ASSET Reuse Library.
Your access privileges will be determined according to the information provided.

Project Name_____

Name of Customer Organization_____

Contract Number_____

Your position on the Contract_____

What type of contract/project are you working on?

___Government ___Private Industry ___Academic Studies

___Army	___Manufacture/Industrial	___Computer Sciences
___Navy	___Military/Government	___Engineering
___Air Force	___Business Services	___Mathematics
___Coast Guard	___Education/R&D	___Other_____
___Marines	___Scientific/Engineering	
___NASA	___Wholesale/Retail	
___DoD	___Transportation/Utility	
___DARPA	___Communications Services	
___State	___Health Care/Hospitals	
___Federal	___Other_____	
___STARS		
___Other_____		

Are you a United States citizen? ___Yes ___No If No, citizenship_____

What level are your primary duties in your organization?

___Executive/Company Management	___Administration
___Computer Systems Management	___Faculty
___Engineering/Technical Management	___Graduate Student
___Sr.Engineering/Sr.Technical Staff	___Other_____

Authorization Signature_____

(Signature of your immediate supervisor, contracting officer (COR/COTR) or other person who can authorize your request to access the ASSET Reuse Library.)

_____

(Printed name and title/rank of signature above.)

D. Signature.

The undersigned, in applying for an ASSET account, does hereby understand and agree to the following terms, restrictions, and conditions:

ASSET makes no warranty, express or implied, as to the accuracy, reliability, fitness for any particular purpose, or safety of the reusable assets available through its library.

ASSET reserves the right to revoke access, privileges, or services without notice.

The undersigned understands that the ASSET account is issued for the applicant only and shall not be transferred or shared.

The undersigned shall comply with any use, duplication, and distribution restrictions that apply to assets extracted from the ASSET Reuse Library.

The undersigned shall not use the ASSET account for any malicious, illegal, or unethical purposes.

Signed_____Date_____

[This form may be photocopied.]

ORDER FORM

Requester's name: _____

Company name: _____

Address: _____

City/State/Zip: _____

Phone Number: _____ FAX: _____

Electronic address: _____

Products Requested (use order number & title listed in this catalog).

ORDER NUMBER TITLE

1) _____

2) _____

3) _____

4) _____

5) _____

6) _____

7) _____

8) _____

Send this form to:

ASSET Librarian
2611 Cranberry Square
Morgantown, WV 26505
Phone: (304) 594-1762
FAX: (304) 594-3951
E-mail: librarian@source.asset.com

If this is the first time you have ordered from ASSET you will need to complete a user account application. The account application allows ASSET to verify your distribution status.

DISCLAIMER

ORDER FORM

Requester's name: _____

Company name: _____

Address: _____

City/State/Zip: _____

Phone Number: _____ FAX: _____

Electronic address: _____

Products Requested (use order number & title listed in this catalog).

ORDER NUMBER TITLE

1) _____

2) _____

3) _____

4) _____

5) _____

6) _____

7) _____

8) _____

Send this form to:

ASSET Librarian
2611 Cranberry Square
Morgantown, WV 26505
Phone: (304) 594-1762
FAX: (304) 594-3951
E-mail: librarian@source.asset.com

If this is the first time you have ordered from ASSET you will need to complete a user account application. The account application allows ASSET to verify your distribution status.

DISCLAIMER

Asset Source for Software Engineering Technology

2611 Cranberry Square
Morgantown, WV 26505
Phone: (304) 594-1762
FAX: (304) 594-3951